



WALLAGA UNIVERSITY

SCHOOL OF POSTGRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE

AFAAN OROMO SHORT TEXT CLUSTERING USING TOPIC MODELING IN DEEP
LEARNING APPROACH

BY

DEREJE FUFA WAKJIRA

June, 2025

NEKEMTE, ETHIOPIA



WALLAGA UNIVERSITY

SCHOOL OF POSTGRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE

AFAAN OROMO SHORT TEXT CLUSTERING USING TOPIC MODELING IN DEEP
LEARNING APPROACH

A thesis submitted to the department of computer science in partial fulfillment for the
degree of masters of Science in computer science

Dereje Fufa Wakjira

Name

Signature

Date

Main Advisor: Mr. Wakgari Dibaba (Ass.Prof) _____



WALLAGA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
P.O. Box: 395, Nekemte, Ethiopia.

APPROVAL SHEET FOR SUBMITTING RESEARCH.

As members of the Board of Examining of the Final MSc. thesis open defense, we certify that we have read and evaluated the thesis prepared by Dereje Fufa under the title: “**Afaan Oromo short text clustering using topic modeling in Deep learning approach**” and this thesis be accepted as fulfilling the thesis requirement 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.

Dr. Obsa Gilo

Department Head

Signature

Date

Chairperson

Dr Obsa Gilo

Internal Examiner

Signature

Date

Signature

Date

Dr. Teklu Urgessa

External Examiner

A blue ink signature of Dr. Teklu Urgessa.

Signature

Date

Final Approval and Acceptance

Thesis Approved by

Department PGC

Signature

Date

Dean of College

Signature

Date

Certification of the Final Thesis I hereby certify that all the correction and recommendation suggested by the board of examiners are incorporated into the final thesis entitled “**Afaan Oromo short text clustering using topic modeling in Deep learning approach**” by **Dereje Fufa**.

Dean of SGS

Signature

Date

Dedication

The dedication of this work is for all my family members, especially my Wife Ms. Abebu Yadeta

STATEMENT OF THE AUTHOR

I Mr.Dereje Fufa hereby declare and affirm that the thesis entitled “Afaan Oromo short text clustering using topic modeling in Deep learning approach” is my own work conducted under the supervision of Mr. Wakgari Dibaba (Ass. Professor) for his generosity in sharing his ideas, suggestions, and encouragements. In the preparation, data collecting, data analysis, and conclusion of my thesis, I followed all of the ethical norms of scholarship. All of the scholarly material in the thesis has been acknowledged through citation. All of the original sources have been properly credited and referenced. I further declare that I followed all academic honesty and integrity principles and that no notion, data, fact, or source in my work was distorted, fabricated, or falsified. This thesis is being offered in partial completion of Wallaga University's Post Graduate Studies degree requirements. I further certify that I have not submitted this thesis to any other university for the purpose of receiving an academic degree, diploma, or certificate. I accept that any breach of the following will result in disciplinary action by the University, as well as legal action by the sources who were not properly cited or who did not get adequate authorization when required.

Name: Dereje Fufa Wakjira

Signature: _____

School/Department: Computer Science

College: Engineering and Technology

Date:_____

Acknowledgments

First and foremost, I want to express my gratitude to almighty God, who is both the beginning and the finish of my effort. I give him credit for everything he has accomplished. With great pleasure and deep gratitude, I would like to express my sincere gratitude to my advisor, Mr. Wakgari Dibaba (Ass.Professor), for his critical comments and guidance, for his invaluable guidance and constant encouragement at each and every step of my thesis work, and for always showing great interest in providing timely support and appropriate suggestions. My family members, especially my wife also deserve special recognition for their emotional support and encouragement throughout my academic career. I pray that God would bless them all. I am also appreciative for the kindness of my colleagues and friends for their generosity in sharing their ideas, suggestions, and encouragements. Finally, I want to express my gratitude and appreciation to everyone who contributed to my thesis effort.

Table of Contents

Acknowledgments	I
Table of Contents	II
List of Acronyms	VI
List of Tables.....	VII
List of Figures	VIII
Abstract	IX
Chapter One	1
1. Introduction	1
1.1. Background of the Study	1
1.2. Motivation	4
1.3. Statement of the Problems.....	4
1.4. Research Questions	6
1.5. Objectives	7
1.5.1. General Objective.....	7
1.5.2. Specific Objectives	7
1.6. Scope and Limitations	7
1.7. Significance of the Study.....	7
1.8. Research Methodology	8
1.8.1. Literature Review	8
1.8.2. Data Collection and preparation	9
1.8.3. Tools and Techniques	9

1.8.4. Evaluation Method	10
1.9. Procedure.....	10
1.10. Thesis Organization.....	10
Chapter Two	11
Literature Review	11
2.2. Text Clustering	11
2.3. Text Clustering Approaches	12
2.3.1. Agglomerative vs Divisive	12
2.3.2. Hard and Soft Clustering	13
2.3.3. Documents-based and Keyword-based Clustering	14
2.4. Algorithms of Text Clustering	14
2.5. Neural Networks	15
2.6. Text feature weighting Algorithms	17
2.6.1. Term frequency (TF)	18
2.6.2. Document frequency (DF).....	18
2.6.3. Term frequency-Inverse Document frequency(TF-IDF).....	18
2.6.4. Cosine similarity.....	19
2.7. Clustering Evaluation Metrics.....	19
2.8. Afaan Oromo Language	21
2.8.1. Overview of Oromo language	21
2.8.2. Afaan Oromo Writing System.....	22
2.8.3. Afaan Oromo Punctuation Marks.....	23
2.8.4. Afaan Oromo Sentence Structure	23
Chapter Three	24

Related Works	24
3.1. Introduction	24
3.2. Probability Clustering for Topic Modeling Approach.....	24
3.3. Partition Clustering Approach	25
3.4. Hierarchical clustering Approach.....	26
3.5.Short text clustering Using Convolutional neural networks (CNN).....	27
3.6. Text clustering for non-Ethiopian Languages	28
3.7. Text Clustering For Amharic Language.....	29
3.8. Text Clustering For Afaan Oromo language	29
Chapter Four	31
The proposed Design	31
4.1. Introduction	31
4.2. Data Collection and Preparation.....	32
4.3. Data Preprocessing	32
4.3.1. Tokenization.....	33
4.3.2. Normalization	33
4.3.3. Stop Word removal.....	34
4.4. Feature Extraction	36
4.5. Generate embedding's using RNN Model.....	37
4.6. BERTopic Algorithm.	38
Chapter Five	40
Experiments and Evaluation	40
5.1. Introduction	40
5.2. Data Collection.....	40

5.3. Data Preprocessing	41
5.4. Feature Extraction	43
5.5. Word Embeddings	43
5.6. BERTopic Modeling	44
5.8. Implementation.....	46
5.8.1.Text clustering With RNN.....	48
5.8.2. Generating Topics using BERTopic.....	49
5.8.3. Evaluating BERTopic Model	50
Chapter Six	51
Conclusion and Recommendations	51
6.1. Conclusion.....	51
6.2. Future Works	52
References	53

List of Acronyms

AI.....	Artificial Intelligence
BERT.....	Bidirectional Encoder Representations from Transformers
BOW.....	Bag-of-words
FBC.....	Fana Broadcasting Corporate
GRU.....	Gated Recurrent Unit
LDA.....	Latent Dirichlet Allocation
LSTM.....	Long Short-Term Memory
NLP.....	Natural Language Processing
OBN.....	Oromia Broadcasting Network
RNN	Recurrent Neural Network
TDT.....	Topic detection and tracking
TF-IDF.....	Term Frequency Inverse Document Frequency

List of Tables

TABLE 1. THE RESEARCH GAP COMPARING WITH (GEMEDA, 2019) OF JIMMA UNIVERSITY	8
TABLE 2.1 <i>ALGORITHM TO REMOVE EXTRANEIOUS CHARACTERS</i>	35
TABLE 5.1 <i>COLLECTED CORPUS WITH ITS SOURCES</i>	40

List of Figures

<i>FIG2.1 AGGLOMERATIVE CLUSTERING</i>	<i>13</i>
<i>FIG2.2. DIVISIVE CLUSTERING (FROM THE NET).....</i>	<i>13</i>
<i>FIG. 2.3 NEURAL NETWORK AND DEEP LEARNING</i>	<i>16</i>
<i>FIG: 4.1 THE PROPOSED DESIGN.....</i>	<i>31</i>
<i>FIG 5.1. COLLECTED DATA BY GRAPH</i>	<i>41</i>
<i>FIG:5.2. CLEANED TEXTS USING NLTK.....</i>	<i>42</i>
<i>FIG:5.3. GENERATED EMBEDDING 'S</i>	<i>43</i>
<i>FIG. 5.4. GENERATED EMBEDDING 'S WITH CLUSTERS</i>	<i>46</i>
<i>FIG 5.5. MODEL LOSS AND MODEL ACCURACY</i>	<i>47</i>
<i>FIG 5.6.LOSS AND ACCURACY OVER EPOCHES</i>	<i>48</i>
<i>FIG 5.7. TEXT CLUSTERING USING BERTOPIC</i>	<i>48</i>
<i>FIG 5.8 TOPIC WORD SCORES</i>	<i>49</i>

Abstract

Text clustering groups texts according to a particular characteristic to determine how similar they are to one another. Keyword-based models have been used as a feature recently, such as the TFIDF model for texts. A key-word-based approach is not feasible for short texts due to their briefness. Moreover, it lacks semantic structure, which limits the range of further text analysis. Moreover, it lacks semantic structure, which limits the range of further text analysis. The topic model was developed to determine the probability distributions of subjects across a predefined collection of phrases or language. Unlike the TFIDF, the topic model has a semantic structure of texts. In addition to ids, the topic model can cluster according to the topic of the cluster. In order to use deep learning to uncover hidden subjects in a collection of short texts, we used topic modeling in this thesis. These days, BERTopic Modeling is one of the most popular and often used topic modeling strategies for deep learning. The main goal of this thesis study is to develop a model for short Afaan Oromo text clustering using the topic modeling approach. We used the BERTopic algorithm as a topic modeling tool. BERTopic enhances topic modeling by using topic based embedding to capture the semantic meaning of text and clustering methods to identify logical subjects. Because it offers powerful visualization tools together with dynamic and hierarchical topic modeling capabilities, it is a versatile and effective method for locating and analyzing topics in text data. To improve topic extraction and clustering accuracy, neural word embedding and topic models have been merged. We have concluded that six clusters is the optimal amount for our experiment based on the Average Coherence Method. We employed 4639 Afan Oromo documents in our training experiment, which covered subjects including politics, health, sports, science, technology, and peace. In terms of word embedding as feature extraction, the final result shows that the model's overall accuracy is 79.24%. Last but not least, we evaluated the BERTopic model using the Coherence score, and we received an 81.34% score. This implies: Semantic similarities between words within themes are more substantial and comprehensible.

Key Words: BERTopic Modeling, Text Clustering, Coherence Model, Recurrent Neural Network (RNN)

Chapter One

1. Introduction

1.1. Background of the Study

A subfield of computer science and linguistics that is multidisciplinary is called natural language processing (NLP). Its main goal is to enable computers to understand and work with human language. NLP integrates statistical and machine learning models with computational linguistics, which models human language using rules. Computers and other digital devices can recognize, comprehend, and produce text and speech thanks to these techniques. Convert text between different languages. Answer oral or printed instructions. Identify or verify users by their voice. Write summaries of a lot of text. Evaluate the speech or text's mood or aim. Create material on demand, frequently in real time, such as text or graphics.

NLP, on the other hand, is utilized for verbal or written communication and idea sharing between people. NLP helps computers comprehend words or statements produced by humans by combining linguistics and artificial intelligence (AI). NLP analyses sounds (phonology), word formation (morphology), sentence structure (syntax), and the ontology of words (semantics) by utilizing information from a number of disciplines (Diksha Khurana, 2017).

Today, Social media and the continuous advancement of information technology have resulted in a significant increase in the volume of information that is accessible online in recent years (Jain, 2019). People's main sources of news and up-to-date information are now news organizations' websites and different social media platforms. Due to the ease of access and low cost of desktop, laptop, tablet, and smartphone devices, the number of individuals utilizing the Internet has increased exponentially. As a result, enormous volumes of text data have been produced. Since most of this data is textual, analysts and researchers can find a wealth of information from it.

To glean insight from this vast volume of text data, many machine learning methods and analysis are required. (Jain, 2019). Clustering is a key element of these procedures. When cluster analysis is applied to textual texts, the outcome is text-based document clustering. It uses natural language processing (NLP) and machine learning to interpret and categorize unstructured textual input. Typically, the word sets that describe the subject matter of the document are extracted first.

They are then compared to other terms to see how frequently they appear in the document. After that, clusters of descriptors can be located and automatically tagged. Assigning topics or categories to the documents is the aim of text data clustering. A group of documents relating to a particular concept or topic is called a cluster. (Jain, 2019). Clustering can help with topic detection, classification, and document organization.

However, the amount of news data on the portal is increasing, which also poses challenges for the website. The traditional text categorization methods and approaches are unable to meet the needs of the vast amount of textual data being created nowadays. Consequently, research on text clustering models has gained popularity recently in the field of text mining. (Y. Wang, 2012).

Without supervision or manually supplied labels, text clustering clusters thematically similar material. Text clusters have proven useful in several applications, including corpus summarization, query expansion, visualization, and language modeling for news suggestions. The similarity is computed using a similarity function. Text clustering can take place at different levels within a comparatively big collection of texts. Topics, terms, sentences, paragraphs, and documents can all be grouped together. Clustering is one of the main techniques for organizing documents to enhance retrieval and make browsing easier. For instance, (D. R. Cutting D. R., 1998) used clustering to build a table of contents for a large collection of papers. Additionally, another researcher in (Anick, 1997) builds context-based retrieval systems by utilizing clustering.

One of the most popular and well-studied data mining methods in the area of text data mining is clustering. It can be used in a variety of domains, such as document organization. (D. R. Cutting D. R.-t., 2011), in-text classification (R. Bekkerman, 2001) as well as the display of particular data domains. Clustering is the process of identifying groupings of related documents within a collection of documents. The similarity is computed using a similarity function. Text clustering can take place at different levels within a comparatively big collection of texts. Topics, terms, sentences, paragraphs, and documents can all be grouped together.

In general, clustering methods can be categorized as partitioned or hierarchical. Due to their poor scalability, hierarchical clustering techniques are not advised for huge data sets, like text.

According to (M. Allahyari et, al, 2017) Certain characteristics of text documents, such their high dimensionality from a large vocabulary, non-normal term distribution, and excessive sparsely, are special enough to treat the data independently and create clustering techniques that are only found in text documents.

Several application domains, including document organization, text crawling, text data retrieval, newsgroup filtering and categorization, and TDT (topic detection and tracking), can benefit from the clustering of text data streams. Traditional static text clustering has a number of issues in these application domains where text data is provided as a continuous stream. (C. C. Aggarwal, 2003). Thus, a review of steam clustering methods based on spherical k-means and other methodologies will be beneficial.

Clustering is not the same as automatic classification since in classification, the clusters are found prior to processing, whereas in classification, the classes are known or predetermined. Clustering uses the distribution and type of data to determine cluster membership, as opposed to classification, which uses a training set a collection of data that has been manually labeled to teach the classifier the relationship between objects and classes.

As a result, the organization and accessibility of text pieces should facilitate users' access to information that they may find pertinent or helpful. Text organization can be approached in a variety of ways. Sorting texts into clear categories that readers can understand is a very efficient way to organize this type of material. Sports, health, politics, science, and so on are examples of categories that classify things according to shared traits or commonalities.

In recent years, there has been a significant surge in both the quantity of text content produced in different natural languages and the number of individuals utilizing the Internet. Additionally, the number of Afaan Oromo broadcasting communication media has significantly increased.

Most of these mediums provide textual content that is kept, written, and presented in Afaan Oromo. Consequently, a wide variety of short texts in different natural languages are now available online. An efficient clustering technique can be used to organize these short texts into meaningful clusters (groups), which facilitates their use in a variety of application areas, including recommendation systems.

1.2. Motivation

Text clustering is important for natural language processing because it makes a lot of data easier to organize and interpret. In addition to searching for information that is relevant to people, including data analysis, it is used to group related documents. It entails a variety of tasks, including applying clustering algorithms, dimensionality reduction, feature extraction, and text preprocessing. This text clustering is done in many languages in the world including Afan oromo language. But it's done in Latent Dirichlet Allocation (LDA) of Machine learning algorithm.

In this thesis we are going to done text clustering by using deep learning algorithm by using Recurrent Neural Network to extract and implement clustered text. Previous researchers have some drawbacks to implement short text clustering using topic modeling, such as:

- Availability of abundant short texts that becomes input for cluster model building
- Problems or challenges in analyzing and interpreting texts that would be solved by the clustering models.

In order to solve these drawbacks, we are going to use Recurrent Neural Networks (RNNs) with BERTopic model. There are no restrictions on the amount of data that can be used for the task when using this deep learning approach. This eliminates the worry about leaving out crucial terms from the scope. This motivated the researchers to do this thesis.

1.3. Statement of the Problems

One of the Cushitic languages with a rich morphology is Afaan Oromo. The Oromia regional state's working language. It is a member of the Cushitic family and is one of the most widely spoken Afro-Asiatic languages. Mostly inhabited by Oromo people, Oromia is one of the Regional States within the current Federal Government of Ethiopia. The Oromo people's native tongue is the Afaan Oromo language.

One of the most extensively used and spoken languages in Ethiopia is Afaan Oromo (Bacha, 2020). More than 50 million people in Somalia, Kenya, Ethiopia, and Egypt speak Afaan Oromo, he said. The third most spoken language on the continent is this Cushitic language. This language is developed using a

variety of NLP applications, including text summarization, named entity recognition, information retrieval, and question answering.

The process of organizing related documents or text passages into groups or categories is known as text clustering. We can streamline massive amounts of text data, uncover hidden patterns, and extract insightful information with this technique. Short texts (like tweets, search queries, online reviews, etc.) are also grouped into various groups without any label information. An unsupervised machine learning method called text clustering groups texts based on their similarity. Similar texts are grouped together, whereas dissimilar texts are grouped together.

Unlike supervised text classification tasks, topic models are an unsupervised learning technique that does not require user-generated labels of training data. Instead, topic models create and, consequently, annotate sizable document collections with thematic data in the form of word clusters called topics. In essence, topic modeling views every single document in a corpus of texts as a model of a bag of words. This indicates that the topic modeling algorithm only considers the frequency of word occurrences and their co-occurrences within each document, ignoring word order and context (Jockers, 2014).

The goal of grouping a collection of texts so that those in the same cluster are more comparable to one another than to those in other clusters is known as document clustering. This is how it differs from topic modeling for document clustering. Text clustering is the process of arranging a collection of texts so that they share characteristics with one another. Text clustering is used in both information retrieval and automated topic extraction.

In this thesis we are going to use RNN algorithm for text clustering in order to cluster text for topic modeling. An outline of how to use an RNN for text clustering is provided here:

- **Text Representation:** Convert text data into numerical form, such as word embedding that can be processed by RNNs.
- **Feature Extraction:** Train the RNN to process the text and extract features that capture the semantic meaning of the texts.
- **Clustering:** Use the features extracted by the RNN to group the texts using a clustering algorithm. RNNs can be used for feature extraction in text clustering; the actual clustering part is typically performed by traditional clustering algorithms.

BERTopic is also used as a pre-trained tool which is a topic modeling technique that leverages BERT and TF-IDF to create dense clusters allowing for easily interpretable topics while keeping important words in the topic descriptions. It supports various topic modeling techniques including guided, supervised, semi-supervised, unsupervised and more. The main reason why we are choosing BERTopic is that Latent Dirichlet Allocation (LDA) of Machine learning has many drawbacks according to (Andronikou, 2022). The main drawback of LDA is as follows:

- ✓ It has no power to handle large vocabulary sizes is its primary flaw. Executors had to restrict their vocabulary in order to fit and apply a good topic model, according to earlier research. This may have an impact on how well the model performs. The most and least common words are typically removed in order to limit vocabulary; this trimming may exclude important terms from the scope.
- ✓ Each document is represented by a bag-of-words model (BOW), and documents are thought of as a probabilistic mixture of latent topics, each of which has a probability distribution over words. Using this method, topic models are sufficient for discovering hidden themes but fail to take into consideration the deeper semantic understanding of a document. A word's semantic representation may be a crucial component of this process.

In order to solve these drawbacks, we are going to use Recurrent Neural Networks (RNNs) with BERTopic model. Using an embedding model, this method can generate relevant subjects that are semantically associated with the text's semantic comprehension. Creating connected and human-understandable themes is crucial when it comes to topic modelling tasks. There are no restrictions on the amount of data that can be used for the task with this deep learning method. By doing this, the worry of eliminating necessary terms from the scope is avoided.

1.4. Research Questions

- What makes topic modeling an effective method for Short Text Clustering?
- Which features are suitable for representing topic model for short texts
- How the suggested models predict text groups?

1.5. Objectives

1.5.1. General Objective

The main goal of this thesis work is to develop a model for short Afaan Oromo text clustering using topic modeling in deep learning approach.

1.5.2. Specific Objectives

The following particular goals are used to accomplish the general objective:

- To study the literature on topic-based short text clustering and the Afaan Oromo language.
- To Collect Afaan Oromo Text corpus for the topic based short text clustering
- To Design a suitable Research process for clustering short texts in Afaan Oromo.
- To apply preprocessing tasks, to make the collected data ready for training and testing the model.
- To train cluster models using deep learning methods.
- To test and evaluate the model.

1.6. Scope and Limitations

This study aims to explore the advantages of employing topic modeling in a deep learning strategy to cluster short Afaan Oromo texts. Designing and developing a model that might combine short Afaan Oromo texts is the aim of the work. For the sake of this study, we only consider text data documents that consist solely of a string of alphabets devoid of any figures, tables, pictures, or other visual representations.

The focus of this thesis work is limited to collecting brief Afaan Oromo texts from different local news sources and categorizing them using the subject modeling technique BERT. Data that is not in written format, such as animation, photos, or videos, will not be taken into account in the suggested work.

1.7. Significance of the Study

Understanding the contents of a collection of millions of documents is aided by topic modeling.

It is a useful application in many fields where a lot of documents are gathered, such as language modeling, bioinformatics applications, recommendation systems, information retrieval, and visualization. They are effectively employed in many texts as well as scientific publications for

understanding. The subject modeler for Afaan Oromo texts will therefore aid in the language's evolution and be crucial for initiating new research on the language's information filtering, recommendation systems, and document similarity. This study proposes a way to integrate topic modeling techniques for creating more accurate topic models by semantically expressing words using a novel word embedding technique, as opposed to using individual words or a bag of words alone, as utilized in standard topic models.

1.8. Research Methodology

1.8.1. Literature Review

This stage is one of the most crucial for developing a comprehensive grasp of the research field. To achieve its objective, this thesis has considered a number of sources, such as journals, recent research articles, new discoveries, and other materials. Examining the literature helps us understand the methods and approaches being used to solve the problem at hand. We look at different articles from different categories that were created with different approaches. Concern over text clustering papers is growing.

In Ethiopian, topic based short text clustering is done in Amharic language by(Deboch, 2020) in Jimma University. In Afaan Oromo, Statistical Topic Modeling for Afaan Oromo Document Clustering is done at Jimma University by(Gemeda, 2019). However, the proposed method constructs vectors based on the presence or absence of words, which is not suitable for short texts. Using the LDA algorithm, the researchers also employed a machine learning approach.

Table 1. The research gap comparing with (Gemeda, 2019) of Jimma university

<i>Authors</i>	<i>Approach they are used</i>	<i>Description</i>
Fikadu Wayesa Gemeda	Statistical Topic Modeling for Afaan Oromo Document Clustering using Latent Dirichlet Allocation (LDA)	✓ This thesis touches on the statistical methods used in semantic analysis of the textual pieces. ✓ This study introduced a topic model that uses LDA to extract high-quality topics from a collection by combining semantic structure. ✓ The proposed model concentrated on document clustering and is not suitable for short texts.

So, we are going to fill the gap for text based vectors rather than document clustering and also we are going to done this thesis by using Recurrent Neural Networks (RNNs) of deep learning algorithm which is a class of artificial neural networks that are particularly well-suited for sequential data, making them a good choice for text-related tasks. RNNs can be modified for text clustering, an unsupervised learning task that groups a collection of texts so that texts in the same group are more similar to each other than to texts in other groups, even though they are typically employed for text categorization.

1.8.2. Data Collection and preparation

Many regional news websites, including as Fana Broadcasting Corporate (FBC), BBC Afaan Oromo, OMN (Oromia Media Network), VOA Afaan Oromo, and Oromia Broadcasting Network (OBN), provide the data used in this thesis. The main justification for using that news page is the wealth of brief material that is accessible for reading. For experimental purposes, the texts have been grouped into multiple groups based on their category on the websites. The study has been finished using a variety of tools.

1.8.3. Tools and Techniques

During the design phase, suggested models and algorithms will be created. Topic modeling is the main technique used in the proposed work to cluster short Afaan Oromo texts. The ability to recognize and extract patterns and features from data regardless of changes in position, orientation, scale, or translation is provided by the convolutional layers' translation invariant properties. Every topic has certain terms that are relevant to it, and every paper has a diversity of words. The Python package BERT facilitates topic modeling by constructing dense clusters that make topics understandable while maintaining important phrases in topic descriptions, using TF-IDF and a range of embedding approaches. It makes the assumption that comparable word groups will be used in texts with related topics. The first few texts in Afaan Oromo were collected from a variety of local news sources. Preprocessing steps like normalization and stop word removal are completed when data collection is finished. Following preprocessing, we use the RNN model to extract topics from the preprocessed data set. The model is extracted using the Python packages matplotlib, gensim, nltk, and sklearn.

1.8.4. Evaluation Method

Test and evaluation findings will be acquired in order to ascertain whether or not the proposed solution accomplishes its goal. To evaluate the proposed method, we will build the dataset and collect short text under the names of the training categories. One aspect of the data preparation work is getting the words ready to be used as data. Average accuracy and coherence score will be used to evaluate topic modeling in the proposed method.

1.9. Procedure

A topic-based short text clustering is developed using a variety of techniques, procedures, and algorithms. On the other hand, selecting a programming language and creating algorithms constitute the computational work. This will be done specifically using the Recurrent Neural Networks (RNNs) algorithm with BERTopic modeling, which is based on deep learning approaches. By evaluating the retrieval effectiveness in the query expansion of a retrieval system for Afaan Oromo topic based short text clustering, the validity of text clustering using BERTopic modeling is demonstrated.

1.10. Thesis Organization

The remaining thesis material is arranged as follows: The relevant literature on document clustering and related works on short text clustering under various conditions, including language differences and RNN model background knowledge, is reviewed in Chapter 2. We present our suggested clustering model for brief Afaan Oromo text in Chapter 3. We proposed the research design in chapter 4. The experimental results and evaluation of the proposed method are presented in Chapter 5. Lastly, a summary and recommendation round out in chapter 6.

Chapter Two

Literature Review

2.1. Introduction

To comprehend the problem domain of the suggested solution from the background literature and to clearly define the boundaries of this thesis work from the most recent cutting-edge publications, journals, and research works that have been reviewed in the fields of text clustering and related ones. A review has been conducted on various text clustering approaches, algorithms for text clustering and feature weighting, clustering metrics, and clustering assessment techniques. There has been discussion of topic modeling tools and comparisons between them. Lastly, a quick overview of the Afaan Oromo language, spelling, and morph syntax has been covered.

2.2. Text Clustering

Text clustering is the application of cluster analysis to text-based materials. It uses natural language processing (NLP) and machine learning to interpret and categories unstructured textual input. It is an important downstream information management duty. The process of classifying comparable texts into groups is known as text clustering. Text clustering differs from earlier manual tagging of clusters or distinct training methods. It is a tactic for dividing or organizing a given collection of patterns into distinct clusters.

There are many similarities between the documents in a single cluster square and many differences between the documents in multiple cluster squares. Communities that use statistics or pattern recognition (P. Poncelet, 2007) developed most of the initial clump techniques, where the goal was to cluster a minor kind of data instances. Clustering has become a crucial processing task technique in recent years. Many everyday tasks, including unsupervised classification, segmentation, and dissection, are handled by this primary operation. For text data, there are numerous clustering algorithms available. A binary vector is used to represent the text document. As an alternative, we can also employ refined representations, which use weighting techniques like TF.IDF, and word embedding's.

2.3. Text Clustering Approaches

The process of organizing a number of texts into clusters according to how similar they are is known as text clustering. It is simpler to study or comprehend related documents when they are grouped together. Let's examine a few popular methods for text clustering:

2.3.1. Agglomerative vs Divisive

There are two types of clustering calculation categories: level and progressive clustering strategies. Just as a company's divisions can be arranged in a level-one or progressive manner, an archive corpus's clusters can be arranged in a level-one or multiple leveled tree structure.

A nested series of divisions is created by hierarchical clustering techniques, with a single, all-inclusive cluster at the top and a single cluster made up of individual points at the bottom (Steinbach, 2000). An upside-down tree can represent the hierarchical clustering result: the individual documents, or the leaves of the tree, represent the lowest level clusters, while the branches represent the intermediate level. The root of the tree represents the highest level of the cluster. Agglomerative techniques are fairly common; similarity measurement techniques can be applied, and the most common distance calculation is quite simple.

Compared to agglomerative clustering, divisive clustering requires a flat clustering method as a "subroutine" to split each cluster until each piece of data has its own singleton cluster. This makes divisive clustering more complex.

Dividing clustering works better if we don't build a complete hierarchy down to individual data leaves. Since we exhaustively search the $N \times N$ matrix distance matrix for the smallest distance in each of the $N-1$ repetitions, a naive agglomerative clustering has an $O(n^3)$ temporal complexity. We can use a priority queue data structure to reduce this complexity to $O(n^2 \log n)$. With a few more optimizations, it can be lowered to $O(n^2)$. (Varun Kanade, 2019).

In contrast, (Absalom E. Ezugwu, 2022) stated that divisive clustering is linear in the number of patterns and clusters when employing an effective flat algorithm such as K-Means, given a fixed number of top levels. It is also more accurate to use a divisive algorithm. Agglomerative clustering makes decisions by first taking into account the global distribution of data, then taking into account neighbor points or local patterns. There is no turning back these initial choices. On the other hand, when divisive clustering makes top-level partitioning decisions, it considers the global distribution of the data.

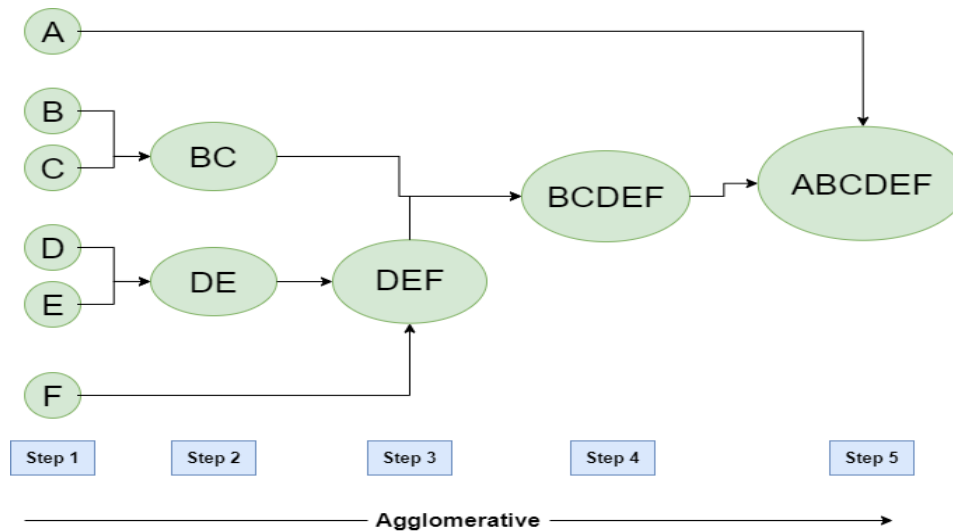


Fig2.1 Agglomerative clustering (from the net)

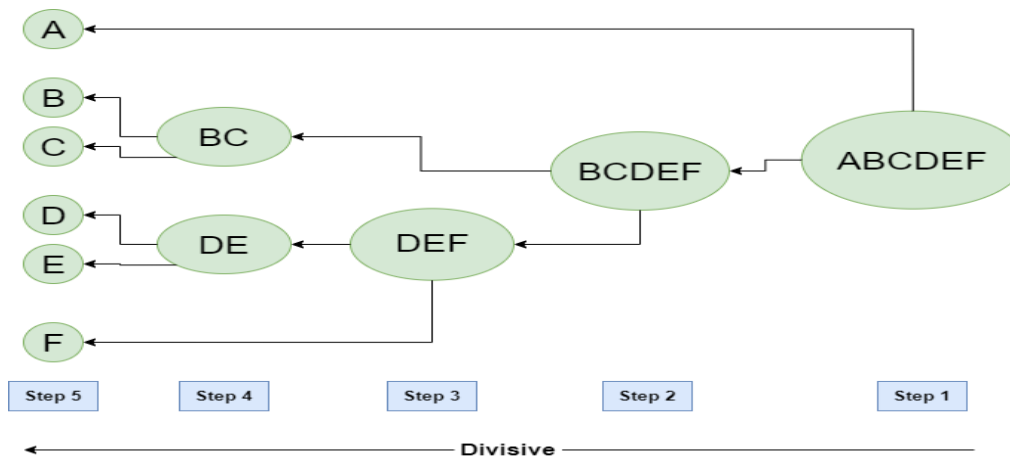


Fig2.2. divisive clustering (from the net)

2.3.2. Hard and Soft Clustering

Clustering methods may produce hard or soft clustering results, depending on whether overlap is tolerated in the final result. It is typical for a single document to cover several topics. It may group into multiple clusters and tag with multiple labels. Under this supposition, collection overlap was permitted. Thus, this type of clustering algorithm that may cluster documents into distinct batches is included in soft clustering. Each item can be a part of multiple clusters, maintaining "soft" boundaries within the collection. Each document that uses soft clustering will typically be assigned to multiple batches(Dhillon, 2003).

But as (K. Mrudula and E. K. Reddy, 2019) points out, there are circumstances in which a single

document is required, and it should only be grouped into the closest category. Because each document in this type of clustering belongs to a single cluster, it is known as "hard clustering." The determination of the most matched cluster is crucial for the hard-clustering algorithms.

2.3.3. Documents-based and Keyword-based Clustering

The features that determine how documents are clustered differ between document-based and keyword-based clustering.

The document vector space model, in which each entry presents the term weighting of a term in the matching document, is subjected to document-based clustering algorithms. As a result, a document is mapped as a data point inside of an extremely high-dimensional space with an axis for each term. It is possible to compute and compare the distances between points in this space. While distant data points are isolated into distinct groups, close data points can combine and cluster into one group. As a result, the related documents were divided or grouped. Since the "document distance" is the foundation of document-based clustering, it is crucial to map the documents into the proper spaces and use the proper distance calculation techniques.

Only particular document features and a restricted set of features are selected by keyword-based clustering algorithms, resulting in the generation of clusters. These few characteristics were chosen because they serve as the fundamental differences between the documents. Similar documents share the same features. Thus, choosing the most fundamental feature is a crucial step in keyword-based clustering.

2.4. Algorithms of Text Clustering

2.4.1. K-means Clustering

K stands for clustering, which separates data points into K groups according to their separation from one another's centers. First, the space's cluster centroid is chosen at random. The distance of each data point from the cluster centroid is then used to assign it to a cluster. New cluster centroids are identified when each point has been assigned to a cluster. This method is repeated iteratively until a good cluster is found. In the analysis, we assume that the number of clusters is fixed and that each group must be given a certain number of points.

When K is not well defined, we have to think about what the optimal value of K should be. K Means is the most effective clustering technique for data that is well-separated. When data points overlap, this grouping should not be used. K Means clustering is faster than other clustering methods. It provides a strong connection between the data points. The quality of a cluster cannot be precisely determined by K Means clusters. Distinct cluster centroid allocations at the start may produce distinct clusters.

2.4.2. Spherical K-means

Taking into account the cosine similarity between the information, spherical k-means is the most popular method for clustering text data (S. Zhong, 2005). During the clustering process, every cluster represents a vector refresh that is, the (standardized) normal of all the text vectors assigned to that group shortly after every report vector has been appointed. The spherical k-means computation appears as follows:

- Give each data point a norm.
- Clustering by locating the center to cluster points that has the lowest cosine angle.
- A basic k-means-like iterative algorithm.

2.5. Neural Networks

Patterns in a set of data are found using algorithms known as neural networks, which are modeled after the structure of the human brain. After processing the input data via the levels of concealment, they output the outcomes. In other words, by mapping inputs to outputs, they learn to approximate a function between any input x and any output y . Additionally, they may be trained on both labeled and unlabeled data; in fact, one of the key benefits of neural networks and other deep learning models is their capacity to conduct unsupervised learning without the use of labeled data. Unsupervised learning is essentially the process by which a model combines comparable things without knowing what those objects are. On the other hand, when it is given data and labels that correspond to the data, it finds correlations between the data and the labels.

For tasks involving clustering, an unsupervised learning method, and classification, a supervised learning task, neural networks can be very effective. Convolutional neural networks, in particular, can be trained to classify images of dogs and cats, for instance. In the training data set, every image is represented by $n \times n$ pixels. The color or darkness of each pixel can then be represented by a number, such as 0 to 255 for color pixels or 0 to 1 for gray scale pixels. Consequently, the input of the neural network will consist of n^2 features in total. Additionally, we must ensure that a category is assigned to every image.

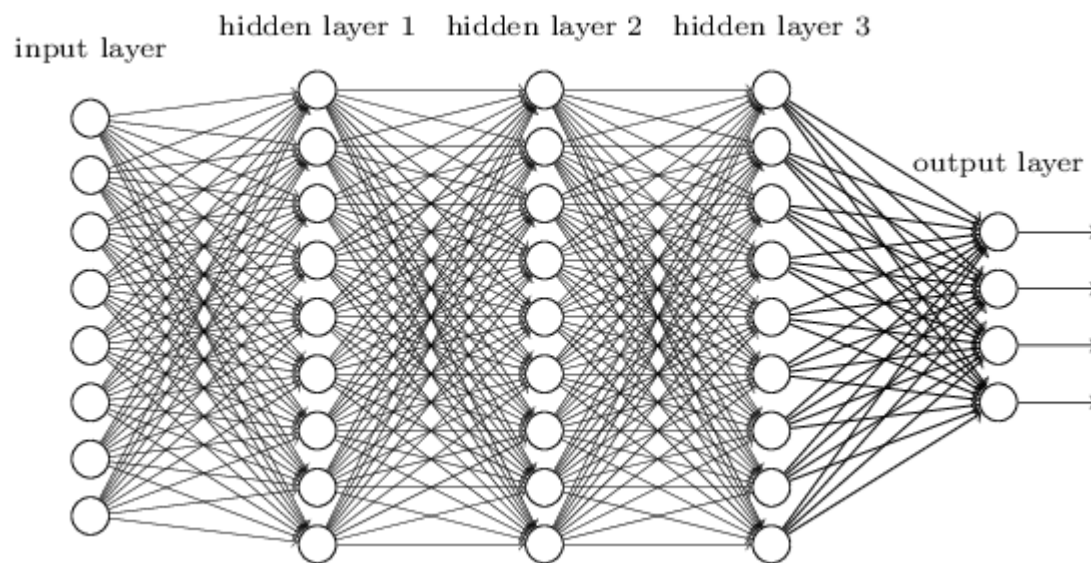


Fig. 2.3 Neural network and deep learning (Michael Nielsen) 2019

Neural networks, however, can also be applied to more common classification issues (like predicting 0 or 1). In these circumstances, the number of columns in the data set would serve as the training data's feature count rather than pixels. The basic components of a neural network, known as layers, combine to examine progressively more extensive characteristics in the training set, enabling the model to recognize patterns. Reshaping the image data from a $n \times n$ array to a $n^2 \times 1$ array is the first layer's responsibility in the image classification scenario. Stated differently, the first layer creates a long line from the rows of pixels in the data. However, for a standard numerical classification, each row of the training data set's information is placed into a $n \times 1$ array, where n is the number of features.

We will want to add fully-connected or densely-connected layers after the initial layer. The training data will be analyzed by these layers for various patterns. Every node in these layers multiplies the input values by certain numbers known as weights, adds the values that enter that node, modifies the outcome

based on the bias of the neuron, and then uses an activation function to normalize the output. The output of the model will be in the final layer. The number of categories that a data point may fall under determines the size of the output array. A 2×1 array.

The output layer when binary classification (1 or 0 / True or False) is used. When categorizing images of people into groups such as newborns, kids, adults, and senior citizens, the result will have a size of 4×1 . The probabilities that a data point falls into a specific category are included in the output. A neural network might determine, for instance, that an image of a cat has a 96% chance of being a cat and a 4% chance of being a dog. If an infant is displayed, the network might forecast that the image has a 94% chance of being an infant, 3% of being a child, 2% of being an adult, and 1% of being an elderly person. When a neural network is used to predict fraud, it can indicate that there is a 92% chance that a specific financial transaction is fraudulent and an 8% chance that it is not.

2.6. Text feature weighting Algorithms

The purpose of document clustering is to group documents into meaningful clusters that accurately represent each document's content. For instance, the news wire needs a lot of human labor to manually assign one or more categories to each document because of the massive volume of text that is uploaded online every day. Effective clustering is therefore crucial. Another problem with document clustering is the vast quantity of terms. In a matrix representation, each term will be a feature and each document an instance. Usually, the number of characteristics and the overall number of words in the dictionary are very comparable. The efficiency of clustering approaches will be much diminished, which poses a considerable issue. However, many of these keywords are stop words that are either unnecessary or irrelevant to the topic. Therefore, dimensionality reduction could be significantly aided by eliminating these unnecessary terms.

In addition to cutting down on processing time, feature selection enhances data interpretability and clustering performance (R. M. P., 2011). When document clustering, a subset of words related to a given cluster will provide more information than all of the words in the documents pertaining to that cluster. Recently, various feature selection techniques, such as term frequency, pruning infrequent terms, pruning highly frequent words, and entropy-based weighting, have been applied to document clustering.

2.6.1. Term frequency (TF)

One of the earliest and most basic term methods that are still very effective is term frequency. It was first mentioned in 1957 (H. P. Luhn, 2010). It is therefore a conventional term selection technique. In a text corpus, papers that are related to the same topic are more likely to use similar phrases. As a result, these frequently used terms will be a trustworthy indicator of a certain topic. A phrase that is commonly used and usually found in a wide range of settings could be said to be uninformative; hence, it is not unselected. It needs to explain this process of getting rid of very popular phrases. Similarly, terms that are highly rare must also be pruned; this is called pruning infrequent terms. Stop words will most likely be trimmed because they are so common.

$$TF(f_i) = \sum_{j \in D_{f_i}} tf_{ij}$$

2.6.2. Document frequency (DF)

It is a helpful method for selecting terms. However, word weighting is inefficient because each selected phrase will be assigned the same weight. Additionally, a document cannot be linked to a TF value. In other words, it cannot distinguish between frequent words that appear in a small subset of documents and common terms that appear in all or most of the documents in the corpus. The frequent words may have discriminative potential for that subset of documents. As an alternative, the term's weight can be scaled using the inverse document frequency (IDF). IDF determines a term's prevalence across all documents:

$$idf(f_i) = \log \frac{|D|}{|D_{f_i}|}$$

where D is the number of documents that contain the term and the total is the number of documents (i.e., sample size).

2.6.3. Term frequency-Inverse Document frequency (TF-IDF)

The TF and IDF measures (above) should now be combined to create a weight for every term in every document. This statistic is called the TF-IDF. It is supplied by

$$tf-idf(f_i, d_i) = tf_{ij} * idf(f_i)$$

TF and IDF assign higher values to phrases that occur frequently in a small number of documents because they have greater discriminating ability. The value of terms that exist in all documents is the lowest, while the value of terms that appear in more documents is the lowest. Superior clustering is more likely for terms with higher TF and IDF scores in document clustering.

2.6.4. Cosine similarity

Regardless of size, two texts are compared for relatedness using a metric known as cosine similitude. The mathematical measure is the cosine of the angle generated by two vectors projected in a multidimensional space. When two texts are represented as term vectors, the correlation between them shows how similar they are. According to(Sharma, 2013), The cosine similarity of two texts on the vector space is a metric that calculates the cosine of the angle between them. It is among the most widely used metrics for text similarity. The following formula can be used to get the cosine of an angle produced by two vectors:

$$similarity = \cos(d_i, d_j) = \frac{D_i \cdot D_j}{\|D_i\| \|D_j\|}$$

When the cosine value is 1, it shows how similar two text documents are, and when it is 0, it shows how distinct they are. In mathematics, the cosine of the angle between two non-zero vectors in an inner product space is measured by a metric called cosine similarity.

2.7. Clustering Evaluation Metrics

The main objective of clustering is to achieve low inter-cluster similarity, or text document similarity between groups, and high intra-cluster similarity, or text document similarity within a cluster. When evaluating a cluster solution, the common metrics of purity, entropy, F-measure, recall, and precision are commonly employed to assess the quality of clusters.(Mangat, 2013). The internal and exterior quality of clustering can also be taken into account. those that are appropriately retrieved are called true positives, while those that are mistakenly obtained are called false positives, according to IR researchers. True negatives are relevant but not recovered, while false negatives are not significant and are not retrieved.

- Precision: the number of positive items predictions that actually belong to the positive class.

$$Precision(p) = \frac{truepositive}{truepositive + falsepositive}$$

(Jayaswal, 2020)

- Recall: refers to proportion number of correctly clustered text over total number of test set.

$$Recall(r) = \frac{truepositive}{truepositive + falsenegative}$$

.(Jayaswal, 2020)

F1-measure: It is an optimization criterion that is used to balance between recall and precision. In other words, it is a harmonic mean of precision and recall.

If the f_1 parameter, which permits recall and precision to be weighted differently, is larger than one, precision takes precedence over recall. Conversely, if it is less than one, recall assumes greater significance. The alternative scenario is that if $=1$, recall and precision are equalized, and the f-measure equation is optimized to:

$$F\text{-measure} = \frac{(f_1 + 1) * precision + recall}{(f_1 * precision) + recall} \dots\dots(Jayaswal, 2020)$$

Where f_1 parameter allows differential weighting of recall and precision, if it is greater than one, then precision becomes more important than recall. On the other hand, if it is less than one, then recall becomes more important. The other possibility is if $=1$ then precision and recall become equal, and the f-measure equation optimized to:

$$F\text{-measure} = \frac{2 * precision * recall}{Precision + recall} \text{ (Jayaswal, 2020)}$$

Topic coherence is a metric that assigns a score based on the degree of semantic similarity between words that score highly in a given topic. The distinction between topics that are statistically artifacts and those that are semantically related is made using the F-measure. Two topics support one another when we say they are coherent. The phrases "football game is a team sport," "football game played

with a ball," and "football game requires great physical effort" are examples of topics that make sense. Let's look at a few of the various coherence scores and how they are determined.

- C_v : measure is based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized mutual information (NPMI) and the cosine similarity.
- C_p : is based on a sliding window, one-preceding segmentation of the top words and the confirmation measure of Fitelson's coherence.
- C_{uci} : measure is based on a sliding window and the point-wise mutual information (PMI) of all word pairs of the given top words.
- C_{umass} : is based on document co-occurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure.

2.8. Afaan Oromo Language

Since this study looks at texts in Afaan Oromo, it is crucial to examine the dialect's linguistic features. Therefore, the structure of the Afaan Oromo language that is believed to be pertinent to the current study was explored in this section.

2.8.1. Overview of Oromo language

As cited by (Moyka, 2020) Afaan Oromo, literally Oromo mouth, is one of the natural languages classified as one of the Cushitic languages spoken in Ethiopia, Somalia, Sudan, Tanzania, and Kenya (D.B.Teferi, 2015). The Cushitic languages are a part of the Afro-Asiatic or Harnito-Semitic language family, which is believed to be divided into six major branches or families. Within the Cushitic branch, there are four more divisions: North, Central, South, and East. The Cushitic languages that are most widely spoken in Ethiopia include Afaan Oromo, Somali, Sidama, Haddiya, and Afar-Sabo. Afaan Oromo is one of the languages spoken by the lowland members of the east Cushitic group. In addition to being a major African language, it is perhaps the third most spoken Afro-Asian language in the world, behind Arabic and Hausa.(Gamta, 1993),(D.B.Teferi, 2015)Afaan Oromo is spoken as a native language (L1-first language) in twelve administrative regional states in Ethiopia, including Oromia

National Regional State. It is taught as a subject in the first (0–4), second (5-8), and third (9–10) cycles of education, and it also functions as a medium of instruction in the first and second cycles. (T.Fufa, 2009). Additionally, it is a subject and a medium of instruction in a number of Ethiopian colleges that have Afaan Oromo departments established to train journalists, teachers, and anybody working in the cultural and public relations sectors..

It is also used in the media, judiciary, government, and places of worship. Outside of its native speakers, many members of Ethiopia's various ethno national groups utilize Afaan Oromo as their lingua franca and daily language..(D.B.Teferi, 2015).

2.8.2. Afaan Oromo Writing System

One common way to visually portray verbal communication is through writing. Qubee, a Latin-derived alphabet, was approved as the official Afaan Oromo writing system.. (Bedane, 2015). Because the letters accurately depict the language's vowels and consonants, this writing system was chosen. The official script for the Afaan Oromo language is the Qubee script, which is derived from Latin. (Abeshu, 2010).

The Afaan Oromo language uses a modified version of the Latin writing system. As a result, the language has many characteristics with English writing, albeit with minor modifications. The "Qubee Afaan Oromoo" writing system, a simple system based on the Latin alphabet, is used to write in Afaan Oromo. The only difference between Oromo and English letters is the writing style. Afaan Oromo is a phonetic language, which means that words are spoken exactly as they are written, according to claims.(Girma D, 2013). contrary to English or other. Here are no hopped or mispronounced noises or scripts in Latin-based languages.

Seven mixed consonant letters known as "Qubee Dachaa" are included in the Qubee Afaan Oromoo writing system, which consists of 26 English letters for a total of 33 letters. Zy, ts, ph, ny, dh, sh, and ch are some of them. "Qubee" has all of the English language's vowels.

A, E, O, U, and "I" are these. In the language, vowels have two unique characteristics that can lead to a range of interpretations. Vowels can be either lengthy or short in nature. If there is only one vowel, it is considered short. The term "long vowel" is used when the number is two, the maximum. Consider

the terms "lafaa" (soft) and "lafa" (ground). In words with doubled consonants, the sounds are more noticeable. For example, damee (branch) or dammee (sweety).

2.8.3. Afaan Oromo Punctuation Marks

Text is punctuated to facilitate reading and make meaning plain. The punctuation marks in Afaan Oromo follow the same pattern as those in English and other languages for the purpose of writing systems, just like in English and other Ethiopian languages. (W.Tesema) For example:

- To separate a list of ideas, concepts, names, items, etc., use the qooduu comma (,).
- The Tuqaa full stop (.) is used in abbreviations and at the conclusion of sentences.
- The MallattooGaafii question mark (?) is used at the end of a direct question or in an interrogative.
- Command and exclamatory sentences are terminated with the Rajeffannoo exclamation mark (!).
- In addition to its many traditional functions, the tuqlamee colon (:): is used to initiate and separate lists, clauses, and quotations..

2.8.4. Afaan Oromo Sentence Structure

(Tamirat, 2017) Sentence structure differs between Afan Oromo and English. Subject-object-verb (SOV) language is used in Afan Oromo. The sequence in which the subject, object, and verb appear is known as SOV. The sentence form known as subject-verb-object (SVO) places the subject first, followed by the verb and the third object. For example, "Mooneeraan bilisa bahe" is an Afan Oromo phrase. "Bahe" is a verb, "bilisa" is an object, and "Mooneeraa" is a subject. It possesses SOV structure as a result. "Mooneeraa has got freedom" is the English version of the sentence, which follows the SVO form. Additionally, the way adjectives are formed in English and Afan Oromo differs. When used in Afan Oromo, adjectives normally come after a noun or pronoun; in English, they typically come before the noun they modify. For example, namicha gaarii (good man) has the adjective gaarii after the noun namicha.

Chapter Three

Related Works

3.1. Introduction

There are many methods that could be employed to accomplish this goal because clustering is a subjective task. Each methodology defines the similarity between data points according to a distinct set of guidelines. Though many clustering algorithms exist, only a small number are widely applied. There are numerous varieties of text clustering algorithms, including probabilistic clustering algorithms, partitioning algorithms, and agglomerative clustering algorithms. The various text clustering research projects using various methodologies are presented in this chapter.

3.2. Probability Clustering for Topic Modeling Approach

One of the most widely used probabilistic clustering algorithms that have received more attention lately is topic modeling. Developing a probabilistic generative model for the corpus of text documents is the primary goal of topic modeling(Blei D, 2001). A topic in topic models is the probability distribution over words, whereas documents are a combination of topics.

Latent Dirichlet Allocation (LDA) (Blei D, 2001) and Probabilistic Latent Semantic Analysis (pLSA)(T. Hofmann, 1999) are the two primary topic models. pLSA for document modeling was introduced. It is challenging to generalize the pLSA model to model new, unseen documents since it lacks a probabilistic model at the document level. (Blei D, 2001)Latent Dirichlet allocation, a versatile generative probabilistic model for discrete data collection, is described by the author. LDA is achieved through a straightforward application of de Finetti's representation theorem, which is based on a basic exchangeability assumption for the words and topics in a document. With appropriate generative probabilistic semantics that make sense for the kind of data it models, LDA can be seen as a dimensionality reduction method in the vein of LSI. Although LDA cannot handle exact inference, it can use a wide range of approximate inference algorithms for inference and parameter estimation. Framework by adding a Dirichlet prior to the mixture weights of topics per document, this model was extended, and it was given the name Latent Dirichlet Allocation(LDA).

Two distinct frameworks were utilized by the authors in (F. Esposito, 2016) to conduct unsupervised topic modeling of the CompWHoB Corpus, a political corpus that compiles transcripts from White

House Press Briefings. They used the LDA model approach, extracting only the topic with the highest probability from each answer/question document, in order to accomplish their goal first. Second, they tested how well dense, high-quality vectors reflected our data by applying the word embedding that is produced from the Word2Vec model to their data. Ultimately, the results have been compared to determine which one performs better on the provided dataset and Findings indicate that word embeddings perform better than the LDA approach, but only when a linguistic task-oriented preprocessing step is completed using purity, 0.54 and 0.46, in that order.

A generative model that combines topic modeling and document clustering has been proposed in work (P. Xie and E. P. Xing, 2013). It has been assumed that, given a corpus, there are multiple latent groups, with each document belonging to a single latent group. A Dirichlet prior expressing preferences over local topics and a set of local topics capturing the particular semantics of documents in each group are present in each group. Furthermore, it has been assumed that a common Dirichlet prior governs the sampling of pro-portion vectors over global topics for all documents, and that a set of global topics shared by all groups capture the common semantics of the entire collection.

In general, the accuracy of clustering methods based on topic modeling, such as LDA + K-Means and LDA + Naïve, is superior to that of factorization- and K-means-based techniques. Their hypothesis that topic modeling can encourage document clustering is supported by this. The topic models' semantic insights can efficiently enable precise similarity measurements, which aid in the creation of coherent clusters.

3.3. Partition Clustering Approach

Partitioned clustering algorithms can either directly or through a series of repeated bisections compute a k-way clustering of a set of documents. A typical computation for a direct k-way clustering is as follows. To start, a group of k documents were chosen from the collection to serve as the k clusters' seeds. Next, each document was assigned to the cluster that corresponded to its most similar seed after its similarity to the k seeds was calculated.

This is what creates the first k-way clustering. The desired clustering criterion function is then optimized by iteratively refining this clustering. The above algorithm is applied recursively to compute 2-way clustering, yielding a k-way partitioning via repeated bisections. The documents were initially divided into two clusters, one of which was then chosen and further divided, and so on. K-1 iterations

of this process result in k clusters. Every one of these bisections is carried out in a way that maximizes the particular criterion function in the resulting two-way clustering solution.

According to the author in (M. Steinbach, 2008), the criterion function does not always lead to the overall k-way clustering solution being at local minima. The technique for choosing which cluster to bisect next is a crucial part of this algorithm. We decided to use the largest cluster in each of our tests since doing so produced clustering solutions that were both balanced and of a respectable quality. Numerous tests, as reported in (Y. Zhao, 2005), demonstrate that clustering solutions derived from repeated bi-sections are on par with or superior to those resulting from direct clustering. Furthermore, because they must solve a straightforward optimization problem at each stage, their computational requirements are significantly lower. For this reason, we compute partition-clustering solutions using this method in all of our experiments.

3.4. Hierarchical clustering Approach

Because they create a collection of clusters that can be described as a hierarchy of clusters, hierarchical clustering algorithms got their name. Both top-down (also known as divisive) and bottom-up hierarchies are possible. Among the distance-based clustering algorithms is the hierarchical clustering algorithm. It gauges the degree of similarity between text documents using a similarity function. a broad synopsis of the text data hierarchical clustering algorithms found in (Murtagh, 1983).

Using the top-down method, we start with a single cluster that contains every document. We divided this cluster into smaller groups iteratively. Every document in the agglomerative method is first viewed as a separate cluster. Then, one cluster eventually included all of the documents that were most similar to one another. Three distinct merging techniques are available for agglomerative algorithms

- The Single Linkage Clustering technique determines the maximum similarity between any two pairs of documents from two groups of documents.
- Group-Average Linkage Clustering: The average similarity between pairs of documents in these groups determines the similarity between two clusters in group-average clustering.
- Complete Linkage Clustering: In this technique, the worst-case similarity between any two documents in these groups is the similarity between two clusters.

Within (Ravi, 2005) the aid of instance and cluster level constraints, documents has been clustered using hierarchical clustering algorithms. There have been must-link and cannot-link constraints associated with their work. They contend that the most effective method for identifying particular cluster types and avoiding others in hierarchical clustering is to employ these constraints. They utilize six real-world UCI datasets to test their state-of-the-art.

The constraint that permits the use of the triangle inequality to reduce computation time has been introduced in an effort to further increase the efficiency of agglomerative clustering. In the work "Evaluation of Hierarchical Clustering Algorithms for Document Datasets"(Karypis, 2002), partitioned algorithms are used to perform comparison analysis and hierarchical clustering has been evaluated for document datasets. Constrained agglomerative algorithms are a recent addition to the class of clustering algorithms. The characteristics of both agglomerative and partitioned algorithms are combined in this algorithm. Twelve datasets were used in the experiment; the largest and smallest datasets contained 4,069 and 878 documents, respectively.

3.5.Short text clustering Using Convolutional neural networks (CNN)

Due to the sparse text representation in short texts, short text clustering is a difficult problem. (Zhang, 2021) We propose a pliable Self-Taught Convolutional neural network framework for Short Text Clustering that can learn non-biased deep text representation in an unsupervised way and flexibly and successfully incorporate more useful semantic features. Within the framework, one known unsupervised dimensionality reduction technique is used to first embed the original raw text features into compact binary codes. Next, to learn deep feature representations, word embeddings are investigated and fed into convolutional neural networks. In the meantime, the output units are used to fit the binary codes that have already been trained during the training phase. Lastly, the researchers cluster the learned representations using *K*-means to obtain the best clusters. Comprehensive testing outcomes show that the suggested tested on three public short text datasets and found that the framework is efficient, adaptable, and performs better than a number of well-known clustering techniques.

3.6. Text clustering for non-Ethiopian Languages

The study using a topic model to group scientific publications is carried out in (C. K. Yau, 2014). The author examines using topic modeling to cluster a number of collections of scholarly papers according to their content. The author gathered a number of scholarly research papers from seven distinct fields in order to assess the suggested methodology. The outcome demonstrates that the topic model outperforms the four topic modeling algorithms.

Using the k-means algorithm and a continuous vector representation of sentences approach, the study in (R. Paredes, 2015) attempts to cluster the Chinese corpus into sentence level. As stated in the paper, clustering is an interesting problem in text clustering where a document might consist of just one sentence. The natural language processing community has been paying particular attention to this problem because it makes it possible to train distinct models for each of the resulting clusters, resulting in models that are more task-focused. Sentence clustering may also be useful for tasks involving statistical machine translation and text recognition, among other NLP applications.

A novel clustering model based on the combination of Word2Vec skip-gram model and Latent Dirichlet Allocation (LDA) was presented by the authors in (etal, 2018). The model creates topic embeddings that contain more information than the BOW model by refining the information of brief texts from academic abstracts based on paragraph features. The probability matrix and word embedding are trained with less data. It has been demonstrated that this approach performs better than certain conventional ones.

Arabic text is attempted to be clustered using an enhanced clustering algorithm and dimensionality reduction in the work in. In order to create a clustering-based classifier for Arabic text documents, this study suggests three different methods: unsupervised, semi-supervised techniques, and semi-supervised with dimension reduction. Following document preparation, stop words are eliminated, and the roots of all terms in each document are obtained. To determine each term's weight in relation to its document, they use a term weighting technique. Next, evaluate each document's similarity to other documents using a similarity measure method. Additionally, calculating accuracy with the support vector machine (SVM), entropy, and F-measure.

Based on semantic enrichment of texts, the authors in (M. Kozlowski, 2019) have presented an approach to cluster small to medium text corpora that contain very short texts. The preprocessing step's

semantic enrichment is a general purpose method. It adds new features (tokens for categories, synsets, glosses, hypernyms, or related words) to the original texts without changing the text clustering algorithms. This methodology can be implemented very easily because the clustering methods still accept texts as input, which can be interpreted as a bag of words. The studies pertaining to semantic enrichment for the data sets demonstrated that, of the methods utilizing Babel Net tools, only the Babel-synsets approach yielded superior outcomes; the other methods did not enhance the quality of clustering.

3.7. Text Clustering For Amharic Language.

(Deboch, 2020) Utilized topic modeling and machine learning to extract latent or hidden topics from a set of brief texts. Latent Dirichlet Allocation (LDA) is currently a well-liked and extensively applied topic modeling technique. Using the LDA library tool, (Deboch, 2020) have implemented the suggested model in Python. The researcher saved the identified topics as features once LDA has extracted the latent and hidden topics from the provided text. The cluster id of the test set text has been determined by calculating the saved feature and test set similarity. They used into the LDA method approach for feature extraction from short Amharic texts, both with and without word embedding. (Deboch, 2020) have gathered a number of brief Amharic texts from various local news agencies' websites that feature various categories in order to assess the outcome. According to the experimental results of the researchers, LDA without word embedding achieves 90% accuracy, while LDA that uses word embedding for feature extraction achieves 97.17% accuracy.

3.8. Text Clustering For Afaan Oromo language

(Gemed, 2019) Of Jimma University proposed Afaan Oromo text clustering by using LDA algorithm. Since the LDA algorithm suffers from the bag-of-model approach, the researcher combined word embedding approach to capture the semantic structure of words and how they are semantically correlated to each other in order to improve the quality of extracted topics. A total of sixteen documents from four distinct categories health, education, sport, and weather conditions were used by them. A clean text corpus and parameter estimation settings that produce interpretable topics are crucial preconditions for obtaining an accurate topic interpretation from the algorithm. The experiments cover every stage required for gathering data, pre-processing, fitting the model, and using document exploration. To estimate the topic and word distributions, they employed Gibbs sampling.

To verify this algorithm's topic extraction effect, experiments are conducted. The LDA used the generated topics to group the documents and conduct exploration. Three evaluation metrics were used in their study: topic coherence, which assesses the coherence of topics, human judgment for topic interpretability, and perplexity, which helps choose optimal LDA model parameters. With ten topics selected from a range of K topics. The researcher estimated an average accuracy of -9.775 for Perplexity. And also scored Topic coherence of PMI at 52.5% and overall human judgement at 66% of F Measure.

However, the proposed method ignores the semantic relationship between words and instead constructs vectors based only on the presence or absence of words, making it unsuitable for short texts. Additionally, the researchers employed the LDA algorithm in a machine learning approach. These methods are impractical for brief texts, as they typically contain a few hundred words or less. This thesis suggests grouping brief texts using a topic modeling technique. Thus, the goal of this thesis work is to use topic modeling, specifically the Deep Learning Algorithm with Neural Network to Cluster Afaan Oromo Short Text, in order to overcome the limitations of the previous work. Additionally, it improves the clustering result by using neural word embedding with BERTopic algorithm for feature extraction, which has not been used in previous works.

Chapter Four

The proposed Design

4.1. Introduction

This chapter provides a brief overview of the suggested architecture and execution of a topic modeling approach for Afaan Oromo short text clustering. Preprocessing, topic modeling, and clustering are the three primary steps in the design and implementation of Afaan Oromo short text clustering. Tokenization, normalization, and stop word removal are examples of preprocessing operations. Convinced steps in topic modeling prepare the texts for topic extraction. Additionally, every action pertaining to the implementation described in this chapter.

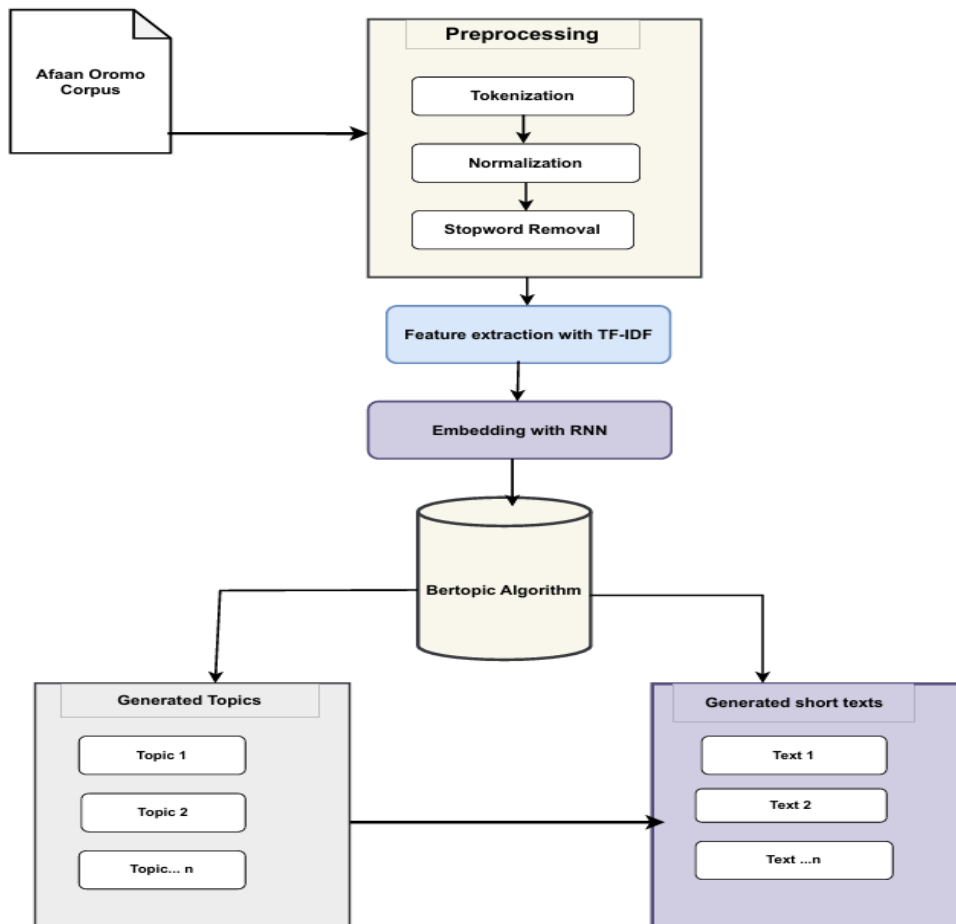


Fig: 4.1 The proposed Design

4.2. Data Collection and Preparation

The process of preparing raw data for further processing and analysis is known as data preparation. Prior to data exploration and visualization, the primary techniques involve gathering, cleaning, and labeling raw data in a format that is appropriate for machine learning algorithms. Before using raw data for business analysis and machine learning, it must be cleaned and combined. This process is called data preparation, or occasionally "pre-processing." Although it might not be the most appealing of tasks, successful data analytics depends on meticulous data preparation. Careful validation, cleaning, and addition are necessary to extract significant and clear concepts from raw data. The strength and validity of the initial information preparation will determine how well any business analysis or model turns out.

To maximize this process, specialized data preparation tools must be used. Assembling all the data required for machine learning is the process of collecting data. Because data can be found in a variety of places, such as devices, laptops, data warehouses, the cloud, and apps. Gathering data can be time consuming. It might be difficult to figure out how to connect to various data sources. With data volumes growing exponentially as well, there is a lot of data to look through.

Additionally, depending on the source, data can have a wide variety of formats and types. Based on this we are going to be collected our data for this thesis from social Medias. These data are categorized in to different topics for text clustering process. So, we are going to classify our data into six main topics such like Sports, Health, technology, Politics, peace and economy.

4.3. Data Preprocessing

Any kind of processing done on raw data to get it ready for another data processing step is referred to as data preprocessing, which is a part of data preparation. It has always been a crucial first step in the data mining procedure. Data preprocessing techniques have been modified more recently to train Artificial Intelligence and machine learning models and to perform inferences against them. Data preprocessing converts the data into a format that can be used for data mining, machine learning, and other data science tasks more quickly and efficiently. To guarantee accurate results, the techniques are typically applied early in the machine learning and artificial intelligence development pipeline.

This module's objective is to produce a set of preprocessed texts, which are necessary for tasks involving clustering. The word frequency in the text and the document's inverse frequency are currently the foundations of the clustering algorithms in use. These methods, however, have drawbacks that make

them unsuitable for this thesis project. Firstly, short texts cannot be clustered using the above methods because there are not enough words to consider, and those texts are very sparse and noisy.

Afaan Oromo text corpus that speeds up and shortens the clustering's runtime. The following actions are part of this language dependent process: tokenization, Normalization and stops word Removal are takes place during preprocessing time.

4.3.1. Tokenization

The process of dividing a given text into units called tokens, or breaking text sentences up into individual words is known as tokenization. Finding the word boundaries that separate two distinct words allows you to do this. Words, numbers, punctuation, special symbols, and so on could be among the tokens. To begin with, each document in the collection would be tokenized into a unique token. This is done in order to eliminate any unwanted reviews from the dataset. The list of tokens is used as input in text mining operations. Tokens were created from strings using a set of delimiters to identify punctuation and numbers.

4.3.2. Normalization

One important phase in natural language processing is normalization. It entails preprocessing and cleaning text data to ensure consistency and suitability for various NLP tasks. Punctuation removal and case normalization.

An essential pre-processing, mapping, and scaling technique that improves the accuracy of forecasting and prediction models is data normalization. This method converts the current data range into a new, standardized range. In order to reconcile disparate prediction and forecasting techniques, normalization plays a crucial role. By standardizing the range of independent variables or features within a dataset, data normalization enhances the consistency and comparability of various predictive models and produces more consistent and reliable results.

Reshaping numerical columns to fit a standard scale is known as normalization, and it is necessary for datasets containing varying units or magnitudes across various features. The basic objective of normalization is to find a common scale for the data while preserving the inherent variances in value ranges.

Rescaling the features to a standard range, usually between 0 and 1, is usually required for this. As an alternative, the features can be changed to have a standard deviation of one and a mean of zero.

The following are some of the reasons why data normalization is necessary:

- Machine learning requires normalization for several reasons. It ensures that each feature contributes equally throughout the learning process, avoiding the overshadowing of smaller features by larger ones.
- It makes optimization algorithms particularly those that rely on gradient descent converge more quickly. Normalization enhances the performance of algorithms that rely on distance, like k-Nearest Neighbors.
- It also addresses model sensitivity issues in algorithms like Support Vector Machines and Neural Networks, improving overall performance. Moreover, because it presumes consistent feature scales, normalization facilitates the application of regularization methods like L1 and L2 regularization.
- Normalization is generally required when working with attributes that have different scales; otherwise, because some attributes have values on a larger scale, the effectiveness of a significant attribute that is equally important (on a lower scale) may be diluted.

4.3.3. Stop Word removal

In natural language processing (NLP), stopwords removal is the process of removing words that are commonly used in a language but have little to no meaning. When dealing with only the most semantically significant words in a text and ignoring words that are rarely semantically relevant, like articles and prepositions, it can be helpful to remove stop words. The process of eliminating stop words from text data is known as "stop words cleaning" in Spark NLP. Stop words are frequently occurring words in a language, such as the, a, and, in, etc., that have little to no meaning and can make it more difficult to complete various natural language processing (NLP) tasks. Eliminating stop words from the text data can greatly increase task accuracy and decrease the amount of computational resources required. Since common words lack semantic meanings, they were eliminated from the corpus to make them unusable for content analysis. These terms are not relevant for content analysis because they dominate the rest of the corpus (Croft, 1998). Terms that were eliminated from the documents that appeared three times or fewer. Actually, we found common words that appeared in a dataset; stop-word lists can vary depending on the domain.

Since they do not provide significant information about the meaning of the target word, the numbers, dates, punctuation, and control characters found in the text of each file were not taken into account when creating the topic model. Words that contained numbers, such as Zd203862 or 2nd, i.e., 2 2~, were removed during the initial stages of preprocessing.

A list of frequently used terms known as "stop words" is typically taken out of texts prior to analysis. Words like "**kana**," "**fi**," "**kanaaf**, **kunneen**, **akkasumas**, **haata'umalee**," and others that don't really convey much information but crop up frequently in texts are included in Afaan Oromo. Because stop word removal helps to reduce the size of text data and facilitates analysis, it is an essential step in text preprocessing. Additionally, it shortens processing times and boosts the effectiveness of algorithms that use text.

Table 4.1. Algorithm to remove Extraneous Characters

<i>Store special characters in temporary variable</i>
<i>Read filename</i>
<i>for each file name in corpus do</i>
<i>Splitintowordandstoreinlist</i>
<i>for Each element of the list</i>
<i>do</i>
<i>ifanelementisoneofthespecialcharacterthen</i>
<i>Discardtheelement</i>
<i>else</i>
<i>Returnelement.</i>
<i>End for</i>

4.4. Feature Extraction

A machine learning technique called feature extraction keeps important or pertinent data while requiring fewer resources for processing. Put differently, feature extraction is the process of creating new features from the original data while preserving its essential information in a more effective way. This results in the conversion of raw data into a set of numerical features that can be easily interpreted and utilized by computer programs.

It is common to come across data with multiple characteristics when working with large datasets, especially in domains like image processing, natural language processing, and signal processing. Many of these characteristics may be redundant or useless. By capturing the key elements of the original data, features are extracted from the data, making processing and analysis of the data more effective.

It is important to extract features for multiple reasons.

- ✓ Lower Computation Cost: Real-world data is typically multifaceted and complex. In the sea of visual data, feature extraction helps us see only the important information. As a result, it simplifies the data, making it easier for machines to handle and process.
- ✓ Enhanced Model Performance: Selecting and extracting important features may reveal details about the underlying mechanisms that produced the data, enhancing the accuracy of the model's performance.
- ✓ Improved Insights: In general, algorithms function better when given fewer features. This is so that the algorithm can focus on the most important features of the data by removing noise and unnecessary information.
- ✓ Preventing Over fitting: Models that have too many features run the risk of becoming over fit to the training set, which impairs their ability to generalize to brand-new, untested data. This is avoided by feature extraction, which rationalizes the model.

For feature extraction in this thesis, we employed TF-IDF, or Term Frequency – Inverse Document Frequency. According to (Guifen Zhao, 2018)tf-idf is One of the most crucial methods for information retrieval, which shows how significant a word or phrase is to a particular document. The frequency of the word in the corpus often counteracts the tf-idf value, which rises in proportion to the number of times a word occurs in the document. This helps to account for the fact that some words appear more frequently overall.

Term Frequency and Inverse Document Frequency are the two statistical techniques used by TF-IDF. The total number of times a term t appears in a document in relation to the total number of words in the document is known as term frequency. The word's amount of information is measured by its inverse document frequency. It calculates a word's weight throughout the entire document. IDF displays a word's prevalence or rarity across all documents. The formula for TF-IDF is $tf * idf$.

Furthermore, we utilized the two weighting factors known as Tf-idf, or inverse document frequency, and term frequency (Jeske, 2019). The number of times a word appears in a specific document is referred to as the term frequency. This value is typically normalized by dividing it by the highest term frequency in the provided document. This results in term frequency values between 0 (for words not appearing in the document) and 1 (for the most frequent word in the document). Term frequencies per word and document are calculated.

$$tf(t,d) = \frac{\text{\# of occurrences of term } t \text{ in document } d}{\text{max \# of term occurrences in } d}$$

On the other hand, the inverse document frequency can only be ascertained word-by-word. It displays how frequently a word appears across the board in the corpus. Take the logarithm of this value to find its inverse. Since one's logarithm is zero, its impact is completely negated.

$$idf(t,D) = \ln \left(\frac{\text{\# of documents}}{\text{\# of documents containing word } t} \right)$$

4.5. Generate embedding's using RNN Model.

Recurrent neural networks, or RNNs, are used to process text data sequences and generate fixed-size vector representations, or embedding's, for each input sequence.

A mapping from a discrete, categorical variable to a vector of continuous numbers is called an embedding. Low-dimensional learned continuous vector representations of discrete variables are called embedding's in neural network contexts. Because they can meaningfully represent categories in the transformed space and reduce the dimensionality of categorical variables, neural network embedding's are helpful. (Koehrsen, 2018).

According to (Koehrsen, 2018), we can use a dense, effective representation where similar words have similar encodings to word embedding's. Crucially, we are not required to manually specify this encoding. A dense vector of floating point values is called an embedding; the vector's length is a parameter that you set. They are trainable parameters (weights learned by the model during training, similar to how a model learns weights for a dense layer) as opposed to manually setting the embedding's values. Word embedding's are frequently 8-dimensional (for small datasets) or up to 1024-dimensional (for large datasets). Although it requires more data to learn, a higher dimensional embedding can capture intricate relationships between words.

In this thesis we are going to use Deep Learning, RNN model for handle sequential data, making them suitable for processing text data where the order of words matters. RNNs can generate embedding's (dense vector representations) for text sequences. These embedding's capture the contextual meaning of the text.

A text's words are presented in a specific order. Recurrent Neural Networks (RNNs) are models that consider word order by processing words one after the other as they occur in the text. It was found that RNNs have a limit on how far they can remember the impact of word correlations since they lack internal memory.

This method was first published in and had a big influence on NLP. It entails describing a word with a vector of a given size, typically 100 or 300. A word's vector (i.e., based on words on its left and right - N words on the left, N words on the right, N typically being 5) is determined by its context within a text. Large text corpuses are usually used for training word embedding. Similar techniques like GloVe were trained on Wikipedia, while Word2Vec was trained on millions of Google News articles. In a word cloud, synonyms, or words with similar meanings, are typically found adjacent to one another.

4.6. BERTopic Algorithm.

According to (Distant, 2022)A variety of algorithms are available for topic modeling. Among the most popular are NMF (Non-Negative Matrix Factorization), LDA (Latent Dirichlet Allocation, and LSA (Latent Semantic Analysis). Prior to using these probabilistic and linear-algebraic algorithms, text must be preprocessed. Additionally, we must state clearly how many topics you wish to identify.

However, the primary drawback of these approaches is that they do not make use of transformer power. Pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) were developed as a result of transformers, which are the de

facto standard in contemporary NLP. Using transformers for topic modeling allows building more sophisticated models that can capture semantic similarities between words.

BERTopic uses BERT embedding with the class-based TF-IDF approach to produce topics that are logical and simple to understand, with automatically generated labels to explain them.

(Distante, 2022) Claims that BERTopic consists of three primary phases. Document embedding must be created first. Sentence-transformers models, which are available for both English and multilingual manuscripts, are the default approach. Since the recently generated document embedding are high dimensional, we must first perform dimensionality reduction, which maintains the embedding' local and global structure, before applying a density-based clustering technique to generate our topic clusters and, if practical, identify outliers.

Each topic (cluster of embedding) can be represented by altering the widely used TF-IDF. We compare the significance of words across papers when we apply TF-IDF to a collection of documents. The relevance scores for words within a cluster would be obtained by computing TD-IDF while treating every text in the cluster as a single document. Topic descriptions can be obtained by extracting the most significant terms from each cluster.

In general, the primary benefit of utilizing BERTopic is that; it provides a high-level interface with default parameters that make it simple to get started and create strong models in a short amount of time. However, it also gives us a lot of options to adjust every configuration to suit our needs and make the most possible use of the package.

Chapter Five

Experiments and Evaluation

5.1. Introduction

We have provided an assessment of the suggested framework in this chapter. This part includes an evaluation of the framework to determine whether or not its objectives are met. We have followed a series of steps, which include a number of activities, to carry out the experiment. The methods and outcomes are covered in the pages that follow in this thesis.

5.2. Data Collection

Data is collected from online media sources. The gathered texts fall into a variety of categories we categorized this data in to different parts , including politics, technology, economy, sports, health, and the peace (Dinagdee, Fayyaa, Ispoortii, Nageenya tekinoooloojii fi siyaasa).

Domain	Source	Number of Sentences
Siyaasa	From OBN, BBC, VOAAfaanOromoOfficialfacebook pages	964
Fayyaa	From FBC, BBCAfaanOromoFacebook Page	972
Dinagdee	From BBC Afaan oromo, OBN Cyber media	657
Tekinoooloojii	From VOA Afaan Oromo, BBC Afaan Oromo Facebook	664
Nageenya	OBN, OMN and FBC Afaan Oromo Facebook Page	658
Ispoortii	OBN, FBC Afaan Oromo, BBC Afan Oromo Facebook	724
Total		4639

Table 5.1 Collected Corpus with its sources

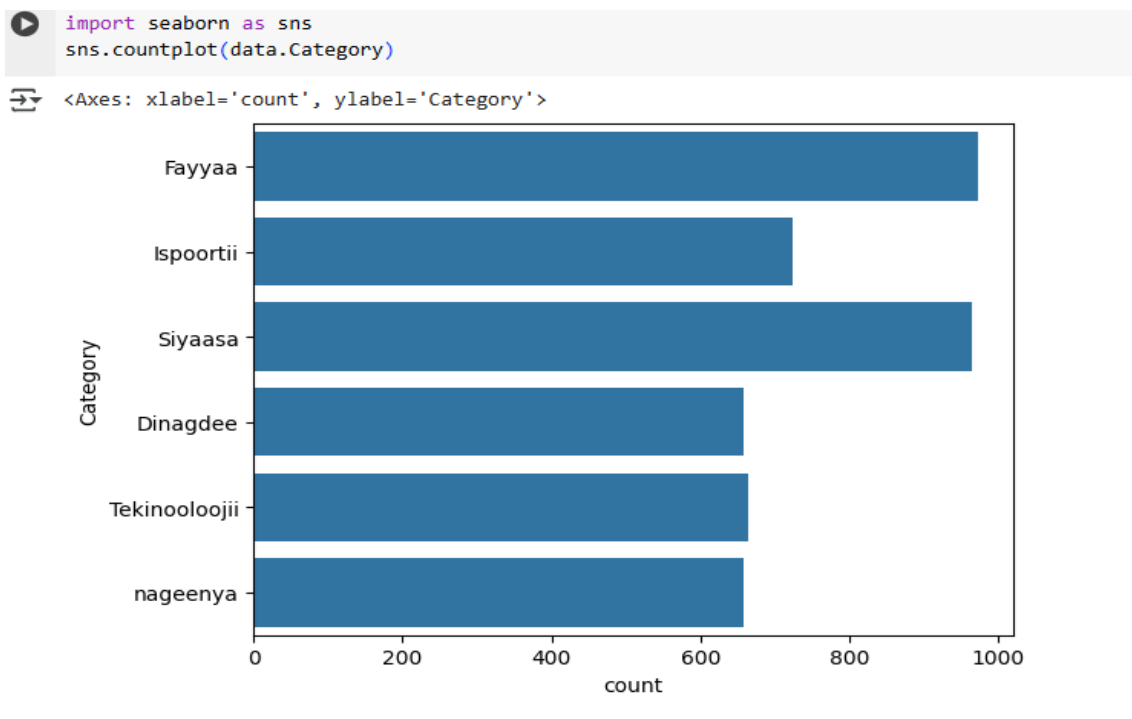


Fig 5.1. Collected data by graph

5.3. Data Preprocessing

Data processing in Python is the process of using Python to manipulate, transform, and analyze data. It includes a number of operations designed to transform unstructured data into structured data or insightful information. Raw data is made suitable for analysis, visualization, and other uses by transforming it into insightful knowledge. Python is a popular option for working with a variety of datasets because it offers a number of libraries and tools that make data processing more efficient.

In this experiment, we use Anaconda software tools for the purpose of programming. Anaconda is a free open-source distribution of programming languages like Python, which is used for predictive analytics, machine learning applications, and data science etc. Anaconda simplifies package management and deployment. From Anaconda supported programming languages, we used python. It is used for general purpose and high-level programming. Python allows you to develop desktop and web applications. We performed our training using word embedding. To implement this research, we also use the following libraries in anaconda prompt.

Pandas

Pandas are a robust and popular Python data analysis and manipulation library. It offers the functions and data structures required to work with big datasets effectively. Among its uses is data manipulation, data cleaning, handling missing values, eliminating duplicates, converting data into a desired format, and conditionally selecting particular rows and columns.

Additionally, it is utilized for data analysis, merging and joining, combining several datasets into one by doing so using a common key, Compute summary statistics, including standard deviation, mean, and median, among others. Calculating statistics for every group by combining data according to categorical variables. This library also performs data visualization by integrating with libraries such as Matplotlib and Seaborn or by using the built-in plotting functions to create different kinds of plots (line plots, bar charts, histograms, etc.)

NLTK

A complete Python library for natural language processing (NLP) applications is called the Natural Language Toolkit (NLTK). It offers resources for working with human language data, facilitating a range of text processing and analysis features. Here are a few typical applications for NLTK Text preprocessing techniques like tokenization are employed with it. Stop word Removal, this thesis's text classification.

```
Category documents \
0 Fayyaa Magaalaa Finfinneetti namoonni kaleen isaanii ...
1 Fayyaa Namni kaleen isaa hojii dhaabde torbanitti yoo...
2 Fayyaa Maallaqa hangasii kaffalanii tajaajila kana ar...
3 Fayyaa Ofii otoo maallaqa itti hin kaffaliin kalee ke...
4 Fayyaa Dhimma kanarratti ispeeshalistii keessoo qaama...
...
4635 nageenya Puutiin waraanicha dhaabuun wareegama lubbuu d...
4636 nageenya Imaammanni alaa bara Bulchiinsa Joo Baayidan, ...
4637 nageenya Bulchiinsi isaanii garuu waraana waggootaaf tu...
4638 nageenya Waraanichi saffisaan akka dhaabatu Ameerikaan ...
4639 nageenya Pireezidaantii Itti Aanaan Ameerikaa Jee Dii V...

cleaned_text
0 magaalaa finfinneetti namoonni kaleen isaanii ...
1 namni kaleen isaa hojii dhaabde torbanitti yoo...
2 maallaqa hangasii kaffalanii tajaajila kana ar...
3 ofii otoo maallaqa itti hin kaffaliin kalee ke...
4 dhimma kanarratti ispeeshalistii keessoo qaama...
...
4635 puutiin waraanicha dhaabuun wareegama lubbuu d...
4636 imaammanni alaa bara bulchiinsa joo baayidan b...
4637 bulchiinsi isaanii garuu waraana waggootaaf tu...
4638 waraanichi saffisaan akka dhaabatu ameerikaan ...
4639 pireezidaantii itti aanaan ameerikaa jee dii v...

[4639 rows x 3 columns]
```

Fig:5.2. Cleaned texts using NLTK

5.4. Feature Extraction

The process of turning unprocessed data into numerical features that machine learning algorithms can use as input is known as feature extraction. In order to simplify the data and highlight the most important information for the learning process, this step is essential. The following are some typical feature extraction strategies and tactics.

Term Frequency-Inverse Document Frequency, or TF-IDF, statistical metric called TF-IDF is used to assess a word's significance in a document in relation to a corpus, or group of documents. It facilitates the conversion of textual material into numerical characteristics. An efficient method for representing text data in a way that highlights the significance of terms inside a document in relation to a collection of documents is to use TF-IDF (Term Frequency-Inverse Document Frequency) in topic modeling. Highlighting key terms in the text and enhancing the interpretability of the identified subjects are two benefits of using TF-IDF in topic modeling. It is possible to efficiently extract significant themes from text data by combining TF-IDF with topic modeling methods.

5.5. Word Embeddings

Word embeddings are essential for improving topic modeling's efficacy and interpretability. Dense vector representations of words that have similar vector representations are called word embeddings. For many natural language processing applications, such as topic modeling, they are useful because they capture the semantic links between words. Compared to more conventional approaches like TF-IDF or Bag of Words, it captures the contextual meaning of words, enabling topic models to comprehend the complex relationships between words.

```
                                embedding
0      [-0.006035092, 0.49043268, 0.0033230695, 0.316...
1      [-0.0030395573, 0.52233934, 0.0010074197, 0.33...
2      [-0.006810121, 0.5409162, 0.004065268, 0.35192...
3      [-0.00783939, 0.49322784, 0.00071601785, 0.316...
4      [-0.00364703, 0.64789504, 0.0040666466, 0.4146...
...
4635  [-0.003612354, 0.4476073, 0.0015585013, 0.2899...
4636  [-0.001522451, 0.4068138, 0.0037941763, 0.2621...
4637  [-0.010162842, 0.7443307, 0.005275308, 0.48112...
4638  [-0.0027431638, 0.48924667, 0.0029820655, 0.31...
4639  [-0.0037189052, 0.2970818, 9.129997e-05, 0.192...

[4639 rows x 5 columns]
```

Fig: 5.3. Generated Embedding's

In order to guarantee that topic models group semantically related words together, synonyms words with similar meanings are positioned closer to one another in the embedding space. It reduces the dimensionality of the data while maintaining semantic information by giving words a more condensed representation. As a result, topic modeling becomes more effective and efficient.

5.6. BERTopic Modeling

According to (Distant, Elisa, 2022) A variety of algorithms are available for subject modeling. Among the most popular are NMF (Non-Negative Matrix Factorization), LDA (Latent Dirichlet Allocation), and LSA (Latent Semantic Analysis). Prior to using these probabilistic and linear-algebraic techniques, text must be preprocessed (stop words, numerals, abbreviations, lemmatization, stemming, etc.). Additionally, we must state clearly how many themes you wish to identify.

However, the primary drawback of these approaches is that they don't make use of transformer power! (And you know, all you need is attention.) Pre trained language models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) were developed as a result of transformers, which are the de facto standard in contemporary natural language processing. Building more complex models that can capture semantic similarities between words is made possible by using transformers for topic modeling.

Topic models can produce more relevant and cogent subjects by utilizing the semantic information that word embeddings gather. This indicates that there is a greater chance that the words in a topic will have a meaningful relationship with one another. By utilizing the semantic information that embeddings capture, BERTopic with embeddings can greatly improve the quality of topic modeling. It is a topic modeling technique that leverages the power of BERT (Bidirectional Encoder Representations from Transformers) embeddings to create coherent and interpretable topics from text data. It works as follows

- **Document Embedding:** BERTopic begins by employing transformer-based language models, such as BERT, that have already been trained to generate document embeddings. The text's semantic meaning is captured by these embeddings.
- **Dimensionality Reduction:** Techniques such as UMAP (Uniform Manifold Approximation and Projection) are used to reduce the high-dimensional embeddings to a lower-dimensional space. This stage facilitates more efficient embedding clustering.

- **Clustering:** Algorithms such as HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) are then used to cluster the reduced embeddings. In this stage, related documents are grouped together according to their embeddings.
- **Class-Based TF-IDF:** BERTopic uses a class-based variant of the Term Frequency-Inverse Document Frequency (TF-IDF) process to extract coherent topic representations. The topics are formed by selecting the most representative words for each cluster.
- **Topic Representation:** The most common and pertinent terms in each cluster serve as the closing topics, offering a concise and logical synopsis of the text's underlying concepts.

BERTopic has been shown to generate coherent topics and remains competitive across various benchmarks involving classical models and more recent clustering approaches to topic modeling. It is particularly useful for exploring and understanding large text datasets, such as news articles, social media posts, or customer reviews.

5.7. BERTopic with RNN

Because of RNNs can process sequential data, they can be used to process text data when word order is important. For text sequences, it can produce embeddings, which are dense vector representations. The context of the text is captured by these embeddings. When combining BERTopic and an RNN, TF-IDF features are generated, embeddings are created from these features using an RNN, and topics are found and visualized using BERTopic. This method makes use of the advantages of topic modeling (for identifying latent topics in text data) and neural networks (for capturing contextual meaning).

In contrast, BERTopic is a topic modeling method that finds topics in text data by using clustering algorithms and transformer-based embedding. It builds dense groups of related documents using embedding, then extracts relevant topics from these groups. We can utilize an RNN to create embedding for our text data and then send these embedding to BERTopic for topic modeling, even though BERTopic mostly requires transformer-based embedding.

In general when we combine RNN model with BerTopic, the role of RNN model is to generate embedding for the text data and to generate embedding as input to BERTopic for topic modeling.

5.8. Implementation.

As we have discussed in above chapters, we have implemented Short text clustering by using topic modeling in deep learning approaches. In this case we used several steps and procedures in this implementations such as data preprocessing, data training, model building and model evaluations.

In these procedures we have gathered 4639 documents from different sources which are classified in six different categories. After preprocessing this all data we used feature extractions to generate the embedding the documents and clustering the documents as we have seen the figure below.

	embedding	cluster
0	[-0.07707214, 0.5183579, -0.012122022, 0.33674...	3
1	[-0.08097517, 0.5538388, -0.015315589, 0.36063...	0
2	[-0.0872481, 0.5826002, -0.013570944, 0.381533...	0
3	[-0.078412674, 0.5194876, -0.015674135, 0.3347...	3
4	[-0.1036306, 0.7085496, -0.019235492, 0.455729...	4
...	...	...
4635	[-0.07151269, 0.4806543, -0.013594917, 0.31291...	3
4636	[-0.06022399, 0.42772278, -0.0090481285, 0.277...	5
4637	[-0.12330375, 0.8085807, -0.01966722, 0.523977...	2
4638	[-0.07675978, 0.53326964, -0.013063049, 0.3444...	3
4639	[-0.0475046, 0.32076147, -0.009971923, 0.20877...	5

Fig. 5.4. Generated embedding's with clusters

As we have discussed above the use of feature extraction is that: A key component of the machine learning pipeline, feature extraction turns unstructured data into useful features that improve the effectiveness, interpretability, and performance of machine learning models. After all feature extractions are done we are going to train the model.

First step we train the RNN (recurrent Neural Network) algorithm because it is designed to handle sequential data and capture temporal dependencies. So we trained the RNN algorithm with our data by using hyper parameters that satisfy with it. We use the following hyper parameters to get the best accuracy of the model

S/no	Hyper parameters	No of Hyper parameters	Results
	Epochs	8	79.24 accuracy
	Batch sizes	2	
	Optimizer	Adam	

```

Epoch 1/8 2320/2320 ----- 55s 22ms/step - accuracy: 0.4983 - loss: 0.7570
Epoch 2/8 2320/2320 ----- 92s 27ms/step - accuracy: 0.5521 - loss: 0.6867
Epoch 3/8 2320/2320 ----- 82s 27ms/step - accuracy: 0.6967 - loss: 0.5869
Epoch 4/8 2320/2320 ----- 79s 26ms/step - accuracy: 0.7574 - loss: 0.5047
Epoch 5/8 2320/2320 ----- 84s 26ms/step - accuracy: 0.7931 - loss: 0.4173
Epoch 6/8 2320/2320 ----- 75s 23ms/step - accuracy: 0.7900 - loss: 0.3782
Epoch 7/8 2320/2320 ----- 79s 22ms/step - accuracy: 0.7874 - loss: 0.3651
Epoch 8/8 2320/2320 ----- 83s 23ms/step - accuracy: 0.7924 - loss: 0.3490

```

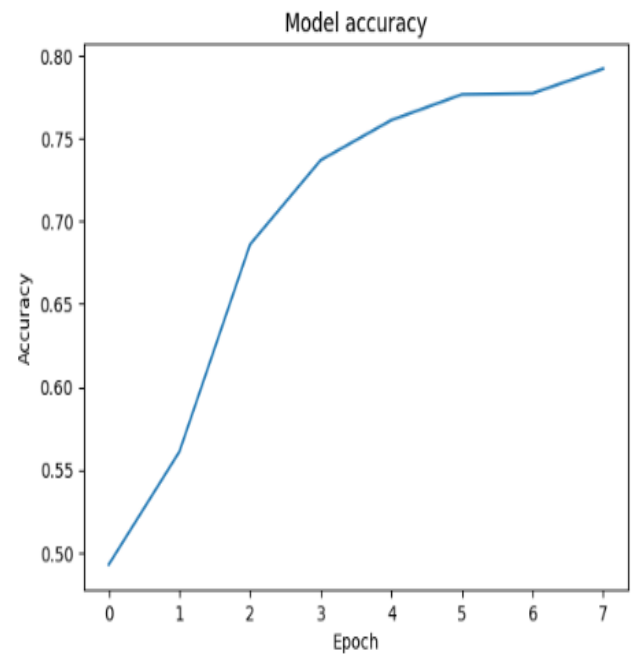
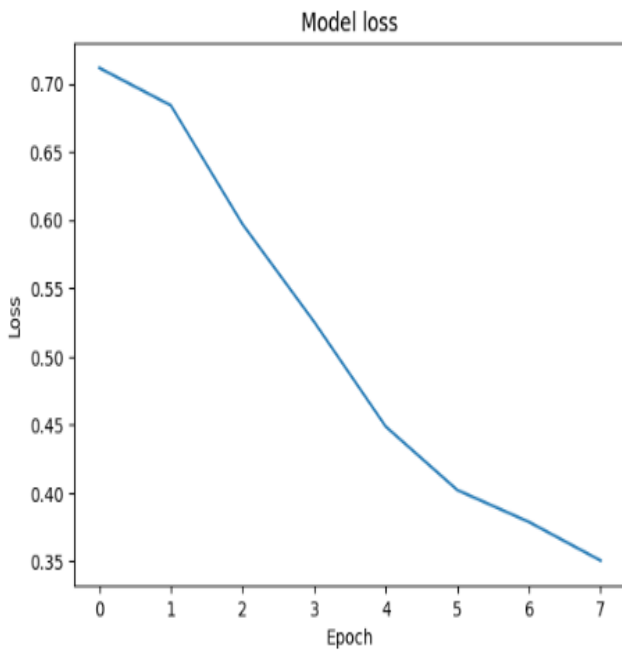


Fig 5.5. Model Loss and Model Accuracy



Fig 5.6. Loss and Accuracy Over epochs

5.8.1. Text clustering With RNN

Using an RNN model for text clustering entails using the sequential nature of RNNs to identify contextual relationships in textual data and then clustering the embeddings that are produced. To generate embeddings, we can preprocess our text data, build and train an RNN model, and then cluster the embeddings using a clustering algorithm. This method makes use of RNNs' sequential structure to identify contextual relationships in text data and produce insightful clusters.

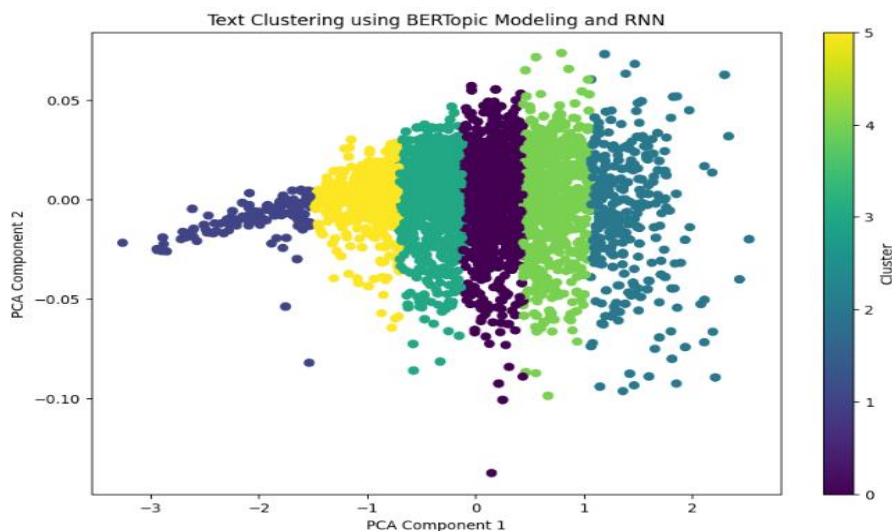


Fig 5.7. Text clustering using BERTopic

5.8.2. Generating Topics using BERTopic

Deep learning models are the main tool used in BERTopic, a machine learning technique. In addition to using deep learning models to create text embedding, the overall strategy also makes use of machine learning methods for topic extraction and clustering. Thus, the best way to characterize BERTopic is as a machine learning method that uses deep learning models to improve its performance.

We can preprocess our text data, create embedding's using a pre-trained model, use BERTopic to cluster the document embedding, extract and visualize the topics, and more by employing this algorithm to extract the topics in the trained data sets. BERTopic creates topics that are meaningful and cohesive by utilizing the semantic information that embedding's capture. For example: here are some top 4 words that the model generates from the topics.



Fig 5.8 Topic word scores

5.8.3. Evaluating BERTopic Model

The quality and coherence of the topics produced by the model are evaluated in order to gauge BERTopic's performance. The semantic similarity of words within a topic is measured by the coherence score. Topics with higher coherence scores are more significant and comprehensible. In topic modeling, the coherence score can be used to gauge how human-interpretable the topics are. The top N words that have the highest likelihood of falling under a given topic are used to represent topics in this instance. In a nutshell, the coherence score calculates the degree of similarity between these words.

In this thesis, we measure BERTopic model by using Coherence score and we have got **81.34%**. This indicates that: the semantic similarities of words with in topics are more significant and comprehensible.

Chapter Six

Conclusion and Recommendations

6.1. Conclusion

We have used topic modeling to cluster short Afaan Oromo texts in this thesis project. Activities such as corpus preparation, preprocessing, design, implementation, and evaluation have been carried out in order to complete the study.

In order to prepare the corpus, various social media platforms were crawled, and various collections of items under various categories were gathered. To improve machine learning, the preprocessing module removes stop words, normalizes the corpus, and tokenizes the news items after they have been collected.

This thesis work's primary task is to use the topic modeling approach to create a model for short Afaan Oromo text clustering. As a topic modeling tool, we employed the BERTopic algorithm. By employing clustering algorithms to find logical topics and transformer-based embedding to capture the semantic meaning of text, BERTopic improves topic modeling. It is a flexible and efficient way to find and analyze topics in text data because it provides dynamic and hierarchical topic modeling capabilities in addition to strong visualization tools. Neural word embedding and topic model have been combined to increase the clustering accuracy and enhance topic extraction.

Using the Average Coherence Method, we have determined that six clusters is the ideal number for our experiment. A total of **4639** brief Afan Oromo documents covering topics such as politics, health, sports, science, technology, and peace were used in our training experiment. The model's overall accuracy with word embedding as feature extraction is **79.24%**, according to the final result. Lastly, we use the Coherence score to measure the BERTopic model, and we obtained an **81.34%** score. This suggests that: words within topics have more significant and understandable semantic similarities.

6.2. Future Works

The following lines of inquiry are suggested in light of the study's findings and closing thoughts.

- A small dataset has been prepared for the BERTopic Model in the current study. Since BERTopic is a good algorithm by nature for analyzing larger datasets, it is necessary to prepare a large and standard dataset for evaluating our approach in order to get better results in topic modeling.
- It is preferable to take into account various feature extraction metrics in order to improve analysis and choose the best data sampling method.
- When dealing with noisy data, like tweets or social media posts, BERTopic may have trouble producing quality topics because of irrelevant information. The next researchers focus on such cases
- Creating a suitable topic model for short texts since they lack sufficient words to be grouped using the bag of words method.

References

- 1) A. Graves. (2014). *Generating Sequences With Recurrent Neural Networks*. 1-43.
- 2) A. Graves, A. M. (2013). *Speech recognition with deep recurrent neural networks*.
- 3) A.Graves, M. S. (2008). *A Novel Connectionist System for Unconstrained Handwriting Recognition*.
- 4) Abera, B. (2020). *Afaan Oromo Word Net*. College of Electrical Engineering and Computing, Ambo University.
- 5) Abeshu, A. (2010). "Automatic Morphological Synthesizer for Afaan Oromoo,".
- 6) Abi, A. (2020). *Semantic Relation Extraction for Amharic Text using Deep Learning Approach*. Addis Ababa.
- 7) Absalom E. Ezugwu, A. A. (2022). *Divisive hierarchical clustering*.
- 8) al., Z. A. (2019). *A State-of-the-Art Survey on Deep Learning Theory and Architectures*. *Electronics*, 1-67.
- 9) Alvarez, J. E. (October, 2017.). Jon Ezeiza Alvarez. *A review of word embedding and document similarity algorithms applied to academic text*. .
- 10) Andronikou, K. (2022). *Topic Modeling with Latent Dirichlet Allocations*.
- 11) Anick, P. G. (1997). *Exploiting clustering and phrases for context-based information retrieval”* *Proceedings of the 20th annual international ACM SIGIR conference on Research and develop- ment in information retrieval* pp. 314—3.
- 12) Asch, V. (2013). *Macro-and micro-averaged evaluation measures*. Belgium.
- 13) Athapornrungkij, P. P. (2005). *Construction of Thai WordNet lexical database from machine readable dictionaries*. In *Conference Proceedings: the tenth Machine Translation Summit, Thailand*, pages 87-92, 2005.
- 14) Bacha, B. A. (2020). *Building WordNet for Afaan Oromoo* vol.11 No 3.
- 15) Baisa, V. a. (n.d.). *Corpus Based Extraction of Hypernym*." *RASLAN*. 2015. p, 69-74.
- 16) Bansal, M. M. (2015). *End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures*.
- 17) Bedane, I. (2015). *The Origin of Afaan Oromo: Mother Language”* .
- 18) Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- 19) Blei D, N. A. (2001). *Latent Dirichlet Allocation (SHORT)*,” *Nips*, vol. 4157698.
- 20) C. C. Aggarwal. (2003). “*A Framework for Diagnosing Changes in Evolving Data Streams*,” *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 575–586, 2003, doi: 10.1145/872824.872826.
- 21) C. Griefenow-Mewis, W. J.G, Möhlig and B. Heine. (2001). *A Grammatical Sketch of Written Oromo*,” *Grammatical Analyses of African Languages*, vol. 16.
- 22) C. Griefenow-Mewis, W. J.G, Möhlig and B. Heine. (2001). *A Grammatical Sketch of Written Oromo*,” *Grammatical Analyses of African Languages*, vol. 16, 2001.
- 23) C. K. Yau, A. P. (2014). *Clustering scientific documents with topic modelin*.
- 24) C.Projection, E. C. (n.d.). *Inducing Information Extraction Systems for New Language Via Cross-Language Projection*.
- 25) Christy, A. K. (2012). *Genetic Algorithm and Confusion Matrix for Document Clustering* . *IJCSI Int J.Comput Sci*.
- 26) Croft, J. M. (1998). , "A language modeling approach to information retrieval," In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*.,
- 27) Cruse, D. A. (2000). *Meaning in Language: An Introduction to Semantics and Pragmatics*, OUP, Oxford, .
- 28) D. R. Cutting, D. R. (1998). *Constant interaction-time scatter/gather browsing of very large document collections*,” *Proc. Annu. Int. ACM SIGIR Conf. Res. Dev. Information Retr.*, pp. 126–134, 1993, doi: 10.1145/160688.160706.

- 29) D. R. Cutting, D. R.-t. (2011). *Browsing A Cluster-based Large Document Approach Collections to Scatter / Gather,* " *Sigir'92*, pp. 318–329, 1992, doi: 10.1145/133160.133214.
- 30) D. Thenmozhi, B. S. (2018). *Deep Learning Approach to Named Entity Recognition and Relation Extraction for Conversational Systems in Indian Languages.*
- 31) D. Zeng, K. L. (2014). *Relation Classification via Convolutional Deep Neural Network*,. th 25th *Interantional Conference on Computational Linguistics*, (pp. 2335-2344).
- 32) D.B.Teferi. (2015). *The Development of Oromo Writing System* . England.
- 33) D.Zeng, K. S. (2014). *Relation Classification via Convolutional Deep Neural Network.*
- 34) Debela Tesfaye and Ermias Abebe. (2010). *Designing a Rule Based Stemmer for Afaan Oromo Text*, *International journal of Computational Linguistics*, Addis Ababa, vol. 1, no. 2,.
- 35) Deboch, K. A. (2020). *Short Amharic Text Clustering Using Topic Modeling.*
- 36) Derry Jatnikaa. (2019). *International Conference on Computer Science and Computational Intelligence* .
- 37) Dhillon, I. S.-d. (2003). "Information-theoretic co-clustering" *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.*
- 38) Diksha Khurana, A. K. (2017).
- 39) Distant, E. (2022). *BERTopic: topic modeling.*
- 40) Distant, Elisa. (2022). *BERTopic: topic modeling.*
- 41) Durme, X. Y. (2014). *Information Extraction Over Structured Data.*
- 42) E. Phaisangittisagul. (2016). *An Analysis of the Regularization between L 2 and Dropout in single Hidden Layer Neural Network*,. *Int. Conf. Intell. Syst. Model Simul.*
- 43) E.Riloff, C. a. (2002). *Inducing Information Extraction Systems for New Language Via Cross-Language Projection.*
- 44) Edgar Altszyler, M. S. (2017). *Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database.* .
- 45) etal, C. L. (2018). *LDA Meets Word2Vec*, " vol. 2, pp. 1699–1706.
- 46) F. Esposito, A. C. (2016). "Proceedings of the Third Ital- ian Conference on Computational Linguistics CLiC-it 2016," *Proc. Third Ital. Conf. Comput. Linguist.* .
- 47) Fadaee, M. H. ((2013)). *Automatic WordNet construction using markov chain monte carlo*. *Polibits* 47.
- 48) Feras Al Tarouti, J. K. (n.d.). *Enhancing Automatic Wordnet Construction Using Word Embeddings.*
- 49) Francis Bond, H. I. (2009). *Enhancing the Japanese WordNet.*
- 50) Fu, R. e. (2014). *Learning semantic hierarchies via word embedding.*" *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- 51) G.E. Hinton, N. S. (2012). *Improving neural networks by prventing co-adaptation of of feature detectors*,. *cs. NE*, 1-18.
- 52) G.G.Chowdhury. (2003). *Natural language processing*, *Annu. Rev. Inf. Sci. Technol.* 37 (1), 51–89.
- 53) Gamta, T. (1993). *Qube Afaan Oromoo: Reason for Choosing the Latin Script for Development an Oromo Alphabet*. *Journal of Oromo Studies*, 36-40.
- 54) Garba, L. (2018). *Synonym Extraction for Afaan Oromoo Words*, *Adama Science and Technology*.
- 55) Gemed, F. w. (2019). *Statistical Topic Modeling for Afaan Oromo Document Clustering using Jimma Univerisity.*
- 56) George A. Miller, R. B. (1995). *WordNet: A Lexical Database for English*. *Communications of the ACM* Vol. 38, No. 11: 39-41.
- 57) Getaneh, M. (2020). *Amharic WordNet construction Using Word Embedding* , *Addis Ababa University* .
- 58) Girma D. (2013). "Afaan Oromo News Text Summarizer,".
- 59) Grefenstette, G. (1993). *Automatic Thesaurus Generation from Raw Text using Knowledge-Poor Techniques. Making Sense of Words*, *9th Annual Conference of the UW Centre for the New OED and Text Research*. .

- 60) Grishman, T. N. (2015). *Relation Extraction: Prespective from Convolutional Neural Networks*. Proc.NAACL-HLT, pp.39-48.
- 61) Guifen Zhao, Y. L. (2018). *TFIDF based Feature Words Extraction and Topic Modeling for Short Text*.
- 62) Gupta, R. C. (2015). *A Simple and Effective Machine Translation Evaluation Metric Based on Recurrent Neural Networks*.
- 63) Guseong-dong, Y. (2002). *Korean Noun Semantic Hierarchy (Wordnet) Construction*.
- 64) H. P. Luhn. (2010). *A Statistical Approach to Mechanized Encoding and Search ing of Literary Information*.
- 65) Harabagiu, B. R. (2010). *UTD:Classifying Semantic Relations by Combining Lexical and Semantic Resources*. *Proceedings of the 5th International Workshop on Semantic Evaluation* , (pp. pp.256-259).
- 66) Hearst, M. A. ((1998)). *Automated discovery of WordNet relations."* *WordNet: an electronic lexical database 2*.
- 67) I. Hendrickx, S. N. (2010). *Multi-Way Classification of Semantic Relations Between Pairs of Nominals*. *Proc. 5th Int. Work. Semant. Eval*, (pp. pp 33-38).
- 68) Idicula, A. A. (2019). *A Relation Extraction System fo Indian Languages*. *Recent Advanced in Engineering Systems*, pp.65-69.
- 69) Idicula, A. A. (2019). *A Relation Extraction System for Indian Languages*. *Recent Advances in Engineering Systems*, (pp. pp 65-69).
- 70) Idicula, A. A. (2019). *A Relation Extraction System for Indian Languages*.
- 71) J, Egger R and Yu. (2022). *A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts*.
- 72) J. Lee, S. S. (2019). *Semantic Relation Classification via Bidirectional LSTM Networks with Entity-Aware Attention Using Latent Entity Typing*.
- 73) J. Pennington R.Socher, a. C. (2014). *Global Vectors for Word Representation*. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* , (pp. pp 1532-1543).
- 74) J., W. (2017). *Recurrent Convolutional Neural Network for Relation Classification*.
- 75) J.Optiz and S.Burst. (2019). *Macro F1 and Macro F1*.
- 76) J.Wen. (2017). *Recurrent Convolutional Neural Network for Relation Classification," Assoc. Adv. Artif. Intell.,.*
- 77) Jain, A. a. (2019). *IoT-based smart doorbell using Raspberry Pi"* *Springer Interna- tional Conference on Advanced Computing Networking and Informatics*.
- 78) Jayaswal, V. (2020). *Confusion matrix, Precision, Recall, and F1 Score*.
- 79) Jeske, S. (2019). *TF-IDF (Term Frequency-Inverse Document Frequency) Explained*.
- 80) Jockers, M. (2014). *Text Analysis with R for Students of Literature*, Springer.
- 81) K. Mrudula and E. K. Reddy. (2019). *"Hard And Fuzzy Clustering Methods: A Comparative Study Hard and Fuzzy Clustering Methods: A Comparative Study*.
- 82) Karani, D. (2018). *Introduction to word Embedding and Word2vec*.
- 83) Karypis, Z. Y. (2002). *Evaluation of Hierarchical Clustering Algorithms for," Perform*.
- 84) Khang Nhut Lam, F. A. (2014). *Automatically constructing wordnet synsets*. In *52nd Annual Meeting of the Association for Computational Linguistics*.
- 85) Koehrsen, W. (2018). *How deep learning can represent War and Peace as a vector*.
- 86) L. Ouyang, H. T. (2019). *Chinese Text Relation Extraction with Multi- instance Multi-label BLSTM Neural Networks*.
- 87) Li Zhang, J. L. (2017). *Automatic Synonym Extraction Using Word2Vec and Spectral Clustering*. *Proceedings of the 36th Chinese Control Conference*. Dalian, China.
- 88) Lin, D. a. (2001). *"Discovery of inference rules for question- answering."* *Natural Language Engineering* 7.4 .

- 89) Luis Espinosa-Anke, S. S. (August 20-26, 2018). *Augmenting Word Embeddings with Unsupervised Relation Vectors*. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2653–2665 Santa Fe, New Mexico, USA, August 20-26, 2018. Mexico, USA.
- 90) M. J. & M. W. Zaki, ". (n.d.).
- 91) M. Kozłowski, H. R. (2019). *Clustering of semantically enriched short texts*, " *J. Intell. Inf. Syst.*, vol. 53.
- 92) M. Mintz, S. B. (2009). Distant supervision for relation extraction without labeled data. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, (pp. pp 1003-1011).
- 93) M. Steinbach, G. K. (2008). "A Comparison of Document Clustering Techniques," *KDD Work*.
- 94) M. Wu, L. W. (2017). *Semantic Relation Classification by Bi-directional LSTM Architecture*. *Adv.Sci.Technol.Lett.ASTL*, , (pp. pp. 205-210).
- 95) M.-F.Moens. (2006). *Information Extraction: Algorithms an Prospects in a Retrieval Context*.Springer.
- 96) M.M.Lopez and J. Kalita. (2017). *Deep Learning applied to NLP*.
- 97) M.Mintz, S. B. (2009). Distant Supervision for relation extraction without labeled data. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, (pp. pp. 1003-1011).
- 98) M.Wu, L. W. (2017). *Semantic Relation Classification by Bi- directional LSTM Architecture*.
- 99) Mangat, S. K. (2013). Evaluation of text document clustering approach based on particle swarm optimization, " *Open Comput. Sci.*, vol. 3, no. 2,.
- 100) Marcelo M. Gomes¹, W. A. (2013). *Automatic Construction of Brazilian Portuguese WordNet*.
- 101) Marrakech, M. S. (2009). *Semi Automatic Construction of a Lexical Ontology for Persian*, 2009, In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.
- 102) Martin, D. J. (2018). *Speech and language Processing* .
- 103) Mikhail Khodak, A. R. (2017). *Automated WordNet Construction Using Word Embeddings*, In *Proc. Workshop on Sense, Concept and Entity Representations and their Applications*.
- 104) Mikolov, K. G. (2013). *Distributed Representations of the Words Phrases and their Compositionality* . cs, CL, PP1-9.
- 105) Mikolov, T. e. (2015). *Computing numeric representations of words in a high-dimensional space*. United States Patent.
- 106) Mooney, [. R. (2005). *A Shortest Path Dependency Kernel for Relation Extraction*. *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.
- 107) Morin, E. a. ((2004)). *Automatic acquisition and expansion of hypernym links*." *Computers and the Humanities* 38.4: 363-396.
- 108) Mortaza Montazery, H. F. (2010). *Automatic Persian WordNet Construction*, in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 846–850.
- 109) Moyka, D. (2020). *Morphological Analyzer for Afaan Oromoo Using Machine Learning*.
- 110) Murtagh, F. (1983). *A survey of recent advances in hierarchical clustering algorithms*.
- 111) N.Chinchor and E.Marsh. (1998). *Information Extraction Task Definition*. MUC-7, (pp. 359-367).
- 112) N.Konstantinova. (n.d.). *Review of Relation Extraction Methods: What is new Out There?* Univ. Wolverhampton,UK.
- 113) P. Poncelet, M. T. (2007). *Data mining patterns: New method and applications*.
- 114) P. Xie and E. P. Xing. (2013). *Integrating document clustering and topic model- ing*, " *Uncertain. Artif. Intell. - Proc. 29th Conf. UAI 2013*, pp. 694–703,.
- 115) P. Zhou, W. S. (2016). *Attention-based bidirectional long short-term memory networks for relation classification*. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, (pp. 207-212).
- 116) Palmer, F. R. (1977.). *Palmer, F. R. Semantics: A New Outline*, CUP, Cambridge. .

- 117) Phye, S. L. (2011). "Construction of Myanmar WordNet lexical database." IEEE Student Conference on Research and Development. IEEE, 2011.
- 118) R. Bekkerman, R. E.-Y. (2001). "On feature distributional clustering for text categorization," SIGIR Forum (ACM Spec. Interest Gr. Inf. Retrieval), pp. 146–153, 2001, doi: 10.1145/383952.383976.
- 119) R. Paredes, J. S. (2015). "Pattern recognition and image analysis: 7th Iberian conference.
- 120) R., M. P. (2011). survey on feature selection in document clustering.
- 121) R., P. a. (2011). Survey on Feature Selection in Document Clustering.
- 122) R.J.Mooney, R. B. (2005). A Shortest Path Dependency Kernel for Relation Extraction.
- 123) Ravi, D. a. (2005). Agglomerative hierarchical clustering with constraints: Theoretical and empirical results.
- 124) Richardella, D. Z. (2003). Kernel Methods for Relation Extraction,. J.ofMachine Learn, 1083-1106.
- 125) S. K. Tomas Mikoloy, M. K. (2010). Recurrent neural network based language model. INTERSPEECH, 1045-1048.
- 126) S. Zhang, D. X. (2015). Bidirectional Long Short-Term Memory Networks for Relation Classification . 29th Pacific Asia Conference on Language, Information and Computation , (pp. pp.73-78).
- 127) S. Zhong. (2005). "Efficient online spherical k-means clustering," Proc. Int. Jt. Conf. Neural Networks, vol. 5,.
- 128) S.S. Zhang, K. H. (2019). Evaluation of the Influences of Hyperparameter and L 2 -norm Regularization on ANN Model for MNIST Recognition. Int. Conf. Intell. Comput, Autom.Syst, (pp. 379-386).
- 129) S.Zhang, D. X. (2015). Bidirectional Long Short-Term Memory Networks for Relation Classification.
- 130) Sanger, W. (2017). Retrieved from Wikipedia. [Online] Available at: <http://www.wikipedia.com/>.
- 131) Schmidhuber, S. H. (1997). LONG SHORT-TERM MEMORY . NEURAL COMPUT, 1-32.
- 132) Sharma, P. S. (2013). Text Document Clustering and Similarity Measures.
- 133) Sheena, N. S. (2016). Automatic extraction of hypernym & meronym relations in english sentences using dependency parser." Procedia Computer Science 93 .
- 134) Silveira, N. (2013). Learning to identify antonym.
- 135) Sisay, A. (2020). INFORMATION EXTRACTION MODEL.
- 136) Snow, R. D. (2006). Semantic taxonomy induction from heterogenous evidence." Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association.
- 137) Snow, Rion, Daniel Jurafsky, and Andrew Y. Ng. (2005). Learning syntactic patterns for automatic hypernym discovery.
- 138) Sojka, R. R. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, 5 2010. ELRA.
- 139) Stanojević, M. (2009). Cognitive synonymy: a general overview.
- 140) Stede, M. (January 2014). . The hyperonym problem revisited" . The hyperonym problem revisited: Conceptual and lexical hierarchies in language generation. Association for Computational Linguistics. .
- 141) Steinbach, M. (2000). A Comparison of Document Clustering Techniques.
- 142) T. Hofmann. (1999). Probabilistic latent semantic indexing," Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval.
- 143) T. Mikolov, G. C. (2013). Efficient Estimation of Word Representations in Vector Space.
- 144) T.Fufa. (2009). A Typology of Verb Derivation in Ethiopia Afro-Asiatic Languages. Netherland.

- 145) Tamirat, W. T. (2017). *Investigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods*. *American Journal of Computer Science and Engineering Survey*.
- 146) Varun Kanade, F. M.-T. (2019). *Hierarchical clustering*.
- 147) Vylomova, E. e. (2015). *Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning*.
- 148) W.Tesema. (n.d.). *ComputerInvestigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in ino Ms Word to Support Text Entry and Input Methods*. *American Journal*.
- 149) Wen, J. (2017). *Recurrent Convolutional Neural Network for Relation Classification*.
- 150) X Yao and B.Van Durme. (2014). *Information Extraction over structured Data:Question Answering with Freebase*. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, (pp. PP. 956-966).
- 151) X. Guo, H. U. (2019). *A Single Attention-Based Combination of CNN and RNN for Relation Classification*.
- 152) X. Han, Y. Z. (2020). *An Attention-Based Model Using Character Composition of Entities in Chinese Relation Extraction*.
- 153) Y. Lui, F. W. (2015). *A Dependency-Based Neural Network for Relation Classification* .
- 154) Y. Wang, E. A. (2012). "TM-LDA: Efficient online modeling of latent topic transitions in social media," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 123–131, 2012, doi: 10.1145/2339530.2339552.
- 155) Y. Xu, L. G. (2015). *Classifying Relation Via Long Short Term Memory Network along Shortest Dependency Paths*. *Proceedings of the 2015 Conference on Empercial Methods in Natural Language Processing* , (pp. 1785-1794).
- 156) Y. Zhao, G. K. (2005). *Criterion functions for document cluster- ing*," *Proc. Thirteen. ACM Conf. Inf. Knowl. Manag.* .
- 157) Zahra Mousavi, H. F. (2017). *Persian Wordnet Construction using Supervised Learning*.
- 158) Zhang, K. L. (2021). *Short Text Clustering with a Deep Multi-embedded Self-supervised Model*. .
- 159) Zheng Yu, H. W. (2015.). *Learning term embeddings for hypernymy identification*. In *IJCAI*, pages 1390–1397, .