JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATICS

Short Amharic Text Clustering Using Topic Modeling

Kebede Assefa Deboch

A Thesis Submitted to the School of Graduate Studies of Jimma University, Jimma Institute of Technology, Faculty of Computing and Informatics in Partial Fulfillment for the Degree of Master of Science in Information Technology.

September, 2020
JIMMA, ETHIOPIA

Declaration

This research work is my original work and has not been presented for a degree in any other university.

Declared By:

Kebede Assefa (Student) Signature: ____________ Date: Oct-09-2020

Advisor: Melkamu Beyene (PhD) Signature: ________________ Date: Oct-09-2020

Co-Advisor : Ephrem Tadesse (MSc) Signature: ________ Date: Oct-09-2020

Approval Sheet

This is to certify that the thesis prepared by Kebede Assefa titled: *Short Amharic text clustering using topic modeling* has been read and approved as meeting the requirements of Faculty of Computing and Informatics in partial fulfillment for the degree of Master of Science in Information Technology complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Advisor Name: Melkamu Beyene (PhD)

Signature: ______________ Date:Oct-09-2020

Co-Advisor Name: Ephrem Tadesse (MSc)

Signature: _______ Date:Oct-09-2020

Chair person: Admas Abtew

Signature: ____________________________, Date: ____________________________

External Examiner Name: Kula Kekeba (PhD)

Signature: ____________________ Date: Oct-09-2020

Internal Examiner Name: Teferi Kebebew

Signature: ____________________________, Date: ____________________________

Acknowledgments

በመጀመሪያ ሁሉን ላደረገ ለልዑል እግዚአብሄር ምስጋና ይገባዋል::

I want to express my deepest thanks to Melkamu Beyene (PhD) and Ephrem Tadesse (MSc) for their help, time, patience,comment and unreserved assistance throughout this study. They really were an inspiration for me to proceed whenever I face difficulties and they are easily approachable. This thesis would not have been possible without their constructive comments on every aspect of the study.

My appreciation goes to for all my friends specially Gashaw Demlew (MSc) staff members of faculty of Computing, Jimma Univesity and for every one who help me to accomplish this study through both mentally and knowledgeably!

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

- A: Accuracy
- BOW: Bag-of-Words
- DMP: Dirichlet Process Mixture Model
- IDF:Inverse Document Frequency
- IR:Information Retrieval
- KNN:K-Nearest Neighbors
- LDA:Latent Dirichlet Allocation
- LSA:Latent Semantic Analysis
- NB:Naive Bayes
- P:Precision
- pLSA:Probabilistic Latent Semantic Analysis
- R:Recall
- SVM:Support Vector Machine
- TF-IDF:Term Frequency - Inverse Document Frequency

Abstract

Text clustering is to group texts according to a certain feature defined on texts to measure the similarity between two texts. keyword-based models like TFIDF of a model for texts have been used as a feature in recent works. Key-word based approach is not feasible for short text due to the texts have only few words. Not only this but also it lacks semantic structure which limits further analysis of texts. The topic model has been developed to discover probabilistic distributions of topics over some fixed set of keywords/vocabulary. Unlike the TFIDF topic model has a semantic structure of texts. The topic model is able to cluster not only using ids but also the topic of cluster.

In this thesis work, we have used topic modeling to discover latent/hidden topics from a collection of short texts through machine learning. Currently, Latent Dirichlet Allocation (LDA) is a popular and widely used topic modeling approach. We have implemented the proposed model in python with LDA library tool. After LDA find the hidden/latent topics from the given text we have saved the identified topics as feature. The saved feature and test set similarity has been calculated to identify the cluster id of test set text. We have investigated the LDA method approach to cluster short Amharic texts with and without word embedding as feature extraction. To evaluate the result, we have collected several short Amharic texts from different local news agencies' websites that contain different groups of categories. The experimental result shows that LDA without word embedding performs 90% of accuracy while LDA with word embedding as feature extraction has an accuracy of 97.17%.

Keywords: Topic Modeling, Text Clustering, Latent Dirichlet Allocation(LDA)

Chapter 1

Introduction

1.1 Background of the Study

In recent years, with the continuous development of information technology and social media, information over the internet increases explosively [1]. News agency websites, different kinds of social media pages have become the main platform for a human to get news and up to date information. Exponential growth in people using the Internet because of ease of access and economical desktop computers, personal computers, pads, tablets, and smartphones has led to a generation of huge amounts of textual data. This data is mostly in form of text and is a great source of information valuable for researchers and analysts.

Knowledge discovery from this huge text data requires a variety of machine learning processes and different kinds of analysis [1]. Clustering is a major part of these processes. The clustering of text data aims at labeling the documents with topics or categories. A cluster is a group of documents that belong to a similar concept or topic [1]. Clustering is useful for topic detection, categorization, and organization of documents.

However, the news data of the News portal is increasing, which also comes up with some challenges to the site. The traditional text classification methods or approaches have been unable to meet the needs of the current huge amount of textual data development. Therefore, the research on text clustering model is always a hot topic in the field of text mining in recent years [2].

A news text clustering system can handle all text data, and it will make an accurate prediction of the cluster labels. So, automatic text clustering can help to complete the text classification function for news platforms with high efficiency. Clustering can help the companies easily cluster news into different categories and manage their text data for better management.

One of the most popular data mining algorithms which have been widely studied in the context of text data mining is clustering. It has a wide range of application areas such as in-text classification [3] and visualizes particular domain of data [4] and document organization [5]. Clustering is the task of finding groups of similar documents in a collection of documents. The similarity is computed by using a similarity function. Text clustering can be in different levels of a relatively large collection of texts where clusters can be documents, paragraphs, sentences, terms, or topics. Clustering is one of the main techniques used for organizing documents to enhance better retrieval and support browsing. For example [6] has used clustering to produce a table of contents of a large collection of documents. In addition, another researcher in [7] exploits clustering to construct context-based retrieval systems.

Clustering methods broadly can be seen as a partition or hierarchical [8]. Generally, hierarchical clustering techniques do not scale well and are not recommended for huge data like text. Characteristics of text documents like high dimensionality due to a lot of vocabulary; highly sparse; and non-normal distribution of terms are unique enough to treat such data separately and devise clustering techniques specific to it.

[9] Proposed k-means approach when applied to normalized text data could produce concept vectors that summarize the text data very close to the most similar one. The text data normalized such that every document is a unit length vector, making the data space Hyper spherical. This variant of k-means called spherical k-means.

Clustering of text data streams can apply in a number of application areas such as newsgroup filtering and categorizing, text crawling, text data retrieval, document organization, and TDT (topic detection and tracking). In such application areas, text data comes as a continuous stream and this presents many challenges to traditional static text clustering [10]. Hence, steam-clustering techniques based on spherical k-means and others are an interesting field to review.

Clustering is sometimes erroneously referred to as automatic classification; however, this is inaccurate, since the clusters found unknown prior to processing whereas in the case of classification the classes are predefined or well known. In clustering, it is the distribution and the nature of data that will give direction to cluster membership, in opposition to the classification where the classifier learns the relationship between objects and classes from a so-called training set, i.e. a set of data correctly labeled by hand, and then predict the learned behavior on the test set.

Therefore, the organizing and access of text items should provide the user with easy access to the information which might be useful or relevant to the user. There are various ways of organizing texts. One of the most successful paradigms to organize such information is classifying texts into different categories that are meaningful to users. Categories signify the organization of items into groups according to their similarities or shared characteristics such as Sport, Health, Politics, science, and so on.

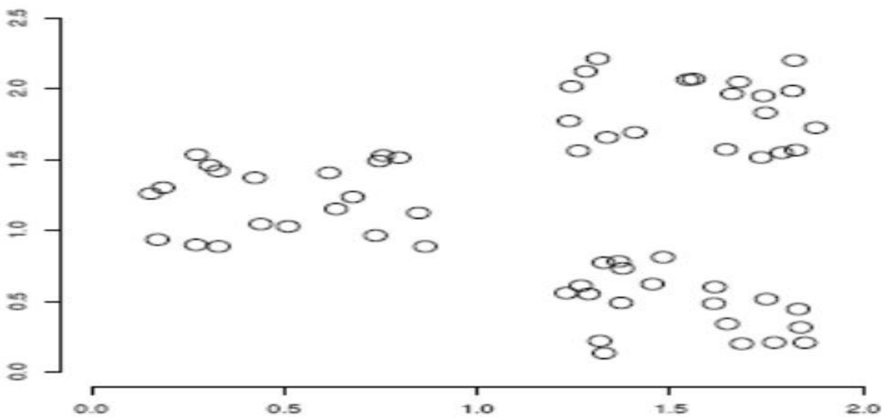

Figure 1.1: An example of a data set with a clear cluster structure [52]

In recent years, Internet users and increasing of text documents written in different natural languages have been dramatic. For example, in year 2016, the number of Internet users in Ethiopia were 4.3% of the population [53]. But, in 2017 the number increased to 11.1%. Additionally, the number of broadcasting communication media in Ethiopia dramatically increased in 2017, from two to more than fifteen. Most of these media are producing text data written, stored and presented using Amharic. Thus, nowadays large collections of short text written in different natural languages are found on the web. Using a good clustering method, these short text can be organized into meaningful clusters (groups), which facilitate using short texts in different application area like recommendation system.

1.2 Motivation

Assume a tourist decision-making problem. How can a tourist decide the place to be visited? A tourist must know about the background information of the places. One way of getting the information is by accessing someone tweet or post who visit the places before. Clustering these tweets or posts based on their area, topics or content will help the users to quickly get access to their required place.

Additionally, short text clustering plays an important role in constructing recommender systems. A recommender system aims to provide users with personalized online product or service recommendations to handle the increasing online information overload problem and improve customer relationship management [11]. Recommender systems utilized in a variety of areas and are most commonly recognized as playlist generators for video and music services like YouTube; product recommenders for online markets; or content recommenders for social media platforms such as Facebook and Twitter; and recommender systems for specific topics like restaurants. Again, clustering texts mentioned about those services or products is a better choice to construct a good recommender system.

1.3 Statement of the Problem

Local Amharic text document clustering studied for a number of years using different approaches and methodologies [12, 13]. However, there are still a number of issues to be addressed. The main issue of the previous studies depends only on keywords to categorize documents to a certain category. The major problem with this approach is that it ignores the semantic relationship between the document's content and the designated category. Categorizing the text document collection using the topic of each document will make the process of clustering more accurate. As opposed to keyword-based technique, this approach guarantees a robust cluster as it is not influenced by word variations.

Previous work [12] on local Amharic text document clustering use bag of words approach. Their work utilizes the frequency of appearance of words in documents. The main problem of this approach is it does not consider the semantic related meaning of words. Which means if two documents use a different collection of words to represent the same topic they assigned in a different cluster. Additionally, short texts have only a few hundred words so this approach is not feasible.

The author in [13] uses an ontology to classify local Amharic news documents into a predefined category. The major problem of this approach is that it is usually difficult to design an ontology that can cover all the concepts mentioned in a text document collection, especially when the documents to be clustered are from a different domain. Classification of a text document to predefined categories excludes different types of text documents that are unrelated or semantically related to the predefined category of documents.

In short text clustering, clustering is performed using short text data like tweets, Facebook status updates, various news feeds, etc. Short text as its name suggests is a text that contains only a few words; for instance, the length of a text in Twitter is only a few words; Search engine queries are mostly short texts. Those characteristics of short text are a challenge for current clustering algorithms. However, it may prove very helpful in extracting meaningful information if this huge unorganized data may be clustered based on some similarity. Unlike document clustering, the major problem in clustering short text is its sparse feature vector due to the short text noisiness and luck of much keywords to construct the feature space.

This thesis work proposed topic modeling that considers each text as a collection of topics while a topic is a collection of words to cluster short texts with better accuracy. The topic model also considers the semantic related meaning of words. Therefore, this thesis work attempt to overcome the limitation of the above work with topic modeling LDA (Latent Dirichlet Allocation) cluster short Amharic texts. The main reason we have used LDA is that this topic modeling helps us to consider semantic relation between words and used to reduce the dimension of document representation. Moreover, this thesis work uses neural word embedding (Skip-gram model) for feature extraction to improve the clustering result that has not been used in the previous works.

At the end of this study, the following research questions has been addressed:

- Why topic modeling is a good approach to cluster shot text?
- What are the appropriate model parameters for topic model training?
- To what extent the proposed model can predict text groups according to the expert's judgment?

1.4 Objective

1.4.1 General Objective

The general objective of this thesis work is to design a model for short Amharic text clustering using topic modeling.

1.4.2 Specific Objectives

In order to achieve the General objective, the following specific objectives are set:-

- Investigating short text clustering properties, challenges, and application areas.
- Reviewing related papers to short text clustering and different literature.
- Designing appropriate algorithm that can cluster short Amharic text.
- Implementing the algorithm for the proposed solution.
- Testing and evaluating the proposed work with the previous works.

1.5 Research Methodology

In order to achieve the objective of the study the following research methods has utilized.

A. Literature review

This phase is one of the crucial steps to get a deep understanding of the research area. To achieve the objective of this thesis has considered different resources like journals, recent research papers, new findings, and other documents. A literature review helps us to understand the current approaches and methods used to solve the defined problem. We have reviewed different papers done with different methodology under different categories. Papers on text clustering get more concerned.

B. Design and Implementation

In the design phase, proposed models and algorithms have been designed. The proposed work considers topic modeling as the main approach to cluster short Amharic text. This thesis work uses Latent Dirichlet allocation (LDA) as a tool to extract hidden topic from the given corpus, which means topic modeling.

First short Amharic texts have been collected from different local news agencies. After the collection is complete data preprocessing tasks like stemming, stop word removal, and normalization has done. After preprocessing the data, we have applied LDA over the preprocessed data set so that it can extract topics. Python libraries Gensim, matplotlib, sklearn, and os have beed used.

C. Prototyping

To implement the proposed model the study developed a prototype for training that contains the appropriate components and a detailed explanation of the components has been given.

D. Data Source and Tools

The data source for this thesis work is collected from different local area news website like Fana broadcasting corporate, Walta info, Ethiopian News Agency, etc. [1]. The main reason for choosing that news portal is that they have enough short text to consider and the texts are freely available. For experimental purpose the texts have been clustered into different category based on their category on the sites. To accomplish the study different kinds of tools has been

[1] The collected texts are from local news agencies website archives and I wonder to give them credit for make the news available freely

employed.

E. Evaluation of the Proposed Work

Results from the proposed solution has been tested and evaluated to check whether it meets its goal or not. To evaluate the proposed method, we have collected short text under the categories we have used for training and prepare the dataset. The data preparation activity includes preprocess the texts to make them appropriate date. Evaluation of the proposed method has been done using precision, recall, and average accuracy.

1.6 Scope and Limitation of the study

This research work was conducted to explore the advantage of using topic modeling to cluster short Amharic texts. The scope of the study was to propose and develop a model that can cluster short Amharic text. In this research work freely available short news items have been used for experimental purpose. In this research work, we considered only text data documents that contain sequence of alphabets without any figure, table, images or any pictorial representations. The scope of this thesis work is limited to collecting short Amharic text from different local news agencies and cluster them into different categories using LDA as a topic modeling approach.

The proposed work has not considered:

- News those are not in text form like video, photo, animation.
- Sentiment analysis of the news.
- Short text of Facebook status update like ’ መልካም አዲስ ዓመት’(melikami ādīsi ‘ameti/happy new year).

1.7 Application of Results

Short text clustering used in:

- In Amharic text document search engines to improve efficiency and search results;

- For Amharic text document filtering, pointing to topic-specific processing mechanisms such as Amharic information extraction and machine translation;
- As input for other information management tasks like organizing, structuring processing, controlling, evaluation, and reporting of information activities;
- For big Amharic document data analytics and recommender system;
- For any organization and application developers, those have a large collection of Amharic documents to automatically cluster documents for better management.

1.8 Structure of Rest of the Thesis

The remainder of this thesis work organized as follows: Chapter 2 reviews the related work on the document clustering. In Chapter 3, we introduce related works done in short text clustering using different approaches under different languages and background knowledge of the LDA model. In Chapter 4, we describe our proposed clustering model for short Amharic text. The proposed method experimental results and evaluation present in Chapter 5. Finally, chapter 6 concludes by summarization and recommendation.

Chapter 2

Literature Review

2.1 Introduction

To understand the problem domain of the proposed solution from the literature background and to identify the clear boundary of this thesis work from the current state-of-art different books, journals, and research works, which related to text clustering and related fields have reviewed. Different text clustering approaches, algorithms for text clustering and feature weighting, clustering metrics, and clustering evaluation methods have been reviewed. Topic modeling tools and their comparisons have been discussed. Finally, a brief introduction about Amharic language, orthography, Amharic morphosyntactic has been discussed.

2.2 Text clustering

Cluster analysis is one of the foremost necessary data processing strategies. It is a central downstream task in information management. Text clustering is that the act of grouping similar texts into categories. Text clustering is not like a separate training method or manual tagging cluster earlier. It is the strategy of partitioning or grouping a given set of patterns into disjoint clusters. The documents within the same cluster square measure a lot of similarities, whereas the documents in several cluster square measure a lot of dissimilarities. Statistics or pattern recognition communities [14] developed most of the initial clump techniques, where the goal was to cluster a minor kind of data instances. In further recent years, clustering refers to a key technique in processing tasks. This main operation is applied to many common tasks like unsupervised classification, segmentation, and dissection.

Many clustering algorithms are available for text data. The text document is represented as a binary vector. Alternatively, we can also use refined representations, which involves weighting methods such as TF-IDF.

2.3 Text Clustering Approaches

A. Agglomerative vs Divisive

Progressive and level (flat) clustering strategies are two kinds of categories of clustering calculations. Fair as divisions in a company may organized be in a progressive fashion or a level one, clusters of an archive corpus may organized be in a various leveled tree structure or in a level style.

Hierarchical Clustering: Hierarchical clustering techniques produce a nested sequence of divisions, with a single, all-inclusive cluster at the top and a single cluster of individual points at the bottom [15]. The hierarchical clustering result can be an upside-down tree: the root of the tree is the highest level of the cluster, the leaves of the tree are the lowest level clusters, which are the individual documents, and the branches of the tree are the intermediate level in the clustering result.

Agglomerative techniques are relatively common: it is quite straightforward the most common distance calculation, and similarity measurement techniques can apply.

B. Online and offline Clustering

Clustering algorithms grouped into online clustering algorithms and offline clustering algorithms based on when clustering performed[16].

Online clustering algorithms perform document clustering when receiving the request and return the request within a specific period. Online clustering requires fast operations (low complexity) and makes the clustering result up-to-date. Online clustering algorithms applied on small or medium size corpus.

Offline clustering, on the contrary, processes the documents and groups them into relevant clusters before receiving the request. When a request is received, offline clustering algorithms perform a few simple operations and then represent the clustering result. Compared with online clustering, offline clustering performs most of the calculations before receiving the requests. It is relatively complex (high complexity) and can apply to large document corpus. The major disadvantage of offline clustering is that the clustering result is not up-to-date. Sometimes it cannot reflect the fact that if a single document or a few documents added into the corpus before most operations are applied in along period. Online clustering and offline clustering have their different applications:

the former work to group the search results, and the latter is to organize the document corpus.

[17] Proposed an online clustering of text documents using the Dirichlet process mixture model. Every cluster modeled according to a multinomial distribution whose parameter follows a Dirichlet prior. For every arriving point, the cluster to join or to open a new cluster decided through probabilities computed using the Dirichlet process. Whenever a point joins an existing cluster, the model is updated using Bayes rule.

C. Hard and Soft Clustering

Based on whether overlapping tolerated in the clustering result, clustering methods might result in hard clustering results or soft ones. It is common for one document to have multiple topics. It might tag with many labels and groups into more than one cluster. In this assumption, the overlapping of collection allowed. So, soft clustering includes this kind of clustering algorithm that may cluster documents into the different batch. Each item may belong to several clusters and keep the boundaries of the collection "soft". In general, with soft clustering, each document will be assigned to more than one batch[18].

However, as stated in [19] some situations need one document that should only be clustered into the most related category. This kind of clustering is called hard clustering because each document belongs to exactly one cluster. It is very important for the hard-clustering algorithms to decide which cluster is the most matched one.

D. Documents-based and Keyword-based Clustering

Keyword-based and document-based clustering is different in the features based on which the documents clustered.

Document-based clustering algorithms are applied to the document vector space model in which every entry presents the term weighting of a term in the matching document. Thereby a document mapped as a data point within an extremely high-dimensional space where each term an axis is. In this space, the distance between points can be calculated and compared. Close data points can merge and cluster into the same group; remote points are isolated into different groups. Thereby the corresponding documents grouped or separated. As document-based clustering is based on the “document distance”, it is very important to map the documents into the right space and apply appropriate distance calculation methods

Keyword-based clustering algorithms only choose specific document features and a limited number of features, the clusters generated. Those limited features are selected because they are the core features between the documents. The features are shared among similar documents. Thereby how to pick up the most core feature is a very important step in keyword-based clustering.

2.4 Algorithms for Text Feature Weighting

Document clustering goal is to isolate documents into meaningful clusters that reflect the content of each document. For example, in the news wire, manually assigning one or more categories for each document requires exhaustive human labor, especially with the huge amount of text uploaded online daily. Thus, efficient clustering is essential. Another problem associated with document clustering is the vast number of terms. In a matrix representation, each term will be a feature and each document is an instance. In typical cases, the number of features will be close to the number of words in the dictionary. This imposes a great challenge for clustering methods where the efficiency will be greatly degraded. However, a huge number of these words stop-words, either irrelevant to the topic, or redundant. Thus, removing these unnecessary words may help significantly reduce dimensionality.

Feature selection not only reduces computational time but also improves clustering results and provides better data interpretability [20]. In document clustering, the set of selected words that are related to a particular cluster will be more informative than the whole set of words in the documents concerning that cluster. Different feature selection methods have been used in document clustering recently, for example, term frequency, pruning infrequent terms, pruning highly frequent words, and entropy-based weighting.

I. Term Frequency

Term Frequency is one of the earliest and most simple yet effective term methods. It dated back to 1957 in [21]. Thus, it is, indeed, a conventional term selection method. In a text corpus, the documents that belong to the same topic more likely will use similar words. Therefore, these frequent terms will be a good indicator of a certain topic. It could be written that a very frequent term that is normally distributed across different topics is non-informative; hence, such a term is not unselected. It has to tell this technique pruning highly frequent terms. Similarly, very rare terms should prune as well and that called pruning infrequent terms. Stop words most likely will prune due to their high frequency. Furthermore, words such as abecedarian will be ignored since they will not be very frequent. TF for a term concerning the whole corpus given by:

$$TF(fi) = \sum_{j \in Dfi} tfij \tag{2.1}$$

II. Document Frequency

TF is an effective term selection method. However, it is not effective in terms of term weighting, where all selected terms will assign the same weight. Also, there is no chance to link TF value to any document. In other words, it cannot distinguish between frequent words that appear in a small set of documents, which could have discriminative power for this set of documents, and frequent words that appear in all or most of the documents in the corpus. In order to scale the term's weight instead, the inverse document frequency (IDF). IDF measures whether the term is frequent or rare across all documents: -

$$idf(fi) = \log \frac{|D|}{|Dfi|} \tag{2.2}$$

Where the total is the number of documents (i.e., sample size) and D is the number of documents that contain the term. The value of IDF will be high for rare terms and low for highly frequent ones.

III. Term Frequency-Inverse Document Frequency

It is time now to combine the above-mentioned measures (i.e., TF and IDF) to produce weight for each term in each document. This measure is called the

TF-IDF. It is given by: -

$$tf - idf(fi, di) = ifij * idf(fi) \tag{2.3}$$

TF and IDF assign greater values to terms that occur frequently in a small set of documents, thus having more discriminative power. This value gets lower when the term occurs in more documents; while the lowest value is given to terms that occur in all documents. In document clustering, TF, and IDF terms that have higher had a higher ability for better clustering.

2.5 Text Similarity Measurement Metrics

Text similarity is all about how close two-piece of texts are both semantically (semantic similarity) and lexically (lexical similarity). Similarity measure plays important role in texts related research and applications like text clustering, topic detection, question answering, and information retrieval. To measure the similarity between two texts researchers found several methods like cosine similarity, Jaccard Similarity, Jensen-Shannon distance, Word Mover Distance, etc.

1) Cosine similarity

Cosine similitude is a measurement used to compare the relatedness of two texts independent of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. When texts are represented as term vectors the similarity of two texts corresponds to the correlation between the vectors. As stated in [22] the cosine similarity of two texts on the vector space is a measure that calculates the cosine of the angle between them. It is one of the most similarity measures used in texts. the cosine of the angle between two vectors is given by:-

$$similarity = \cos(di, dj) = \frac{Di.Dj}{||Di||\,||Dj||} \tag{2.4}$$

When the cosine value is 1 the two text documents are similar, and 0 if there is nothing in common between them. Mathematically speaking, Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

2) Jaccard Similarity

Jaccard similarity or intersection over union is defined as the size of the intersection divided by the size of the union of two sets. In other words, Jaccard similarity looks for the whole weight of shared terms to the total sum of terms that are available in both of the two texts however are not shared terms [22]. The Jaccard similarity for two documents A and B is given by:-

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2.5}$$

The Jaccard similarity is between 0 and 1, 1 means the two texts are the same, and 0 means different. If A and B are both empty, define J(A,B)=1.

3) Euclidean Distance

Euclidean distance calculates the distance between two real-valued vectors. It is the default distance used in classification (K-nearest neighbors), clustering (K-means to find the "k closest points" of a particular sample point). Another example is hierarchical clustering, agglomerative clustering (complete and single linkage) where you want to find the distance between clusters.

It is just a distance measure between a pair of samples p and q in an n-dimensional feature space:

$$\sqrt{\sum_{i=1}^{n} (qi - pi)^2} \tag{2.6}$$

4) Jensen-Shannon distance

Jensen-Shannon distance tells us which documents are statistically close to each other by comparing the difference in their distributions. Jensen-Shannon is a technique for estimating the similarity between two likelihood disseminations.

For two distributions p and q the Jensen-Shannon similarity is given by:

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M) \tag{2.7}$$

Where $M = \frac{1}{2}(P+Q)$

2.6 Text Clustering Algorithms

Before we make go directly into clustering algorithms, let us initially set up certain manners by which we can portray and recognize them. There are a couple of manners by which this is possible:

In hard clustering every item belongs to exactly one cluster whereas in soft clustering items may be in one or more clusters. In hierarchical clustering, items combine hierarchically so they will end in one root. A non-hierarchical approach generates some categories by partitioning a dataset giving a set of non-overlapping groups having no hierarchical relationships between clusters.

A) K-means

K-means is the most known flat clustering algorithm. The objective function of k-means is to minimize the average squared distance of objects from their cluster centers, where a cluster center defined as the mean or centroid μ of the items in a cluster C:

$$\vec{\mu}(C) = \frac{1}{|C|} \sum_{\vec{X} \in C} \vec{X} \tag{2.8}$$

The ideal cluster in K-means is a sphere with the centroid as its center of gravity. Ideally, the clusters should not overlap. A measure of how well the centroids represent the members of their clusters is the Residual Sum of Squares (RSS), the squared distance of each vector from its centroid summed over all vectors.

$$RSSi = \sum_{\vec{X} \in Ci} \left\| \vec{X} - \vec{\mu}(Ci) \right\| 2$$

$$RSS = \sum_{i=1}^{K} RSSi$$

K-means can start with selecting as initial clusters centers K randomly chosen objects, namely the seeds. It then moves the cluster centers around in space to minimize RSS. This is done iteratively by repeating two steps until a stopping criterion is met.

Reassigning objects to the cluster with the closest centroid.

Re-computing each centroid based on the current members of its cluster.

B) Spherical K-means

Spherical k-implies is the most well-known strategy for clustering text data where the calculation takes cosine similarity between information [23]. In the clustering process, each cluster means vector refresh, just after all report vectors have been appointed, as the (standardized) normal of all the text vectors appointed to that group. The spherical k-means calculation looks like:-

- Normalize each data point.
- Clustering by finding center with minimum cosine angle to cluster points.
- Similar iterative algorithm to basic k-means.

2.7 Clustering Evaluation Metrics

The main aim of clustering is reaching high intra-cluster similarity (similarity of text documents within a cluster) and low inter-cluster similarity (similarity of text documents from different groups). When comparing a cluster solution, we can consider the internal and external quality of clustering, the standard measures of Purity, Entropy, F-measure and recall, precision are often commonly used to determine the quality of clusters [24]. In terms of IR Scholars define that values that are correctly retrieved are named true positives while values that are wrongly retrieved are false positive. True negatives are values which are relevant but not retrieved and false negative are not important and not retrieved.

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

$$precision(p) = \frac{true\ positive}{true\ positive + false\ positive} \tag{2.9}$$

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

$$recall\ \ (r) = \frac{true\ positive}{true\ positive + false\ negative} \tag{2.10}$$

c) F-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.

$$F - measure = \frac{(\beta 2 + 1) * precision + recall}{(\beta 2 * precision) + recall} \tag{2.11}$$

Where β 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:

$$F - measure = \frac{2 * precision * rcall}{precision + recall} \tag{2.12}$$

d) Topic coherence: This measure scores a single topic by measuring the degree of semantic similarity between high scoring words in the topic. F-measure is used to distinguish the difference between semantically related and statistically artifacts topics. When we say that two topics are coherent, they support each other. An example of coherent topics 'football game is a team sport', 'football game played with a ball' and 'football game needs great physical effort'. There are different coherence score and let us see some of them how they can be calculated.

- ***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.
- ***C_npmi* :** is an enhanced version of the C_uci coherence using the normalized point-wise mutual information (NPMI).

- ***C_a***: is based on a context window, a pairwise comparison of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity.

e) Purity: according to [24], Purity is an external evaluation criterion of cluster quality. It is the percent of the total number of objects (data points) that were classified correctly, in the unit range [0...1].

$$purity = \frac{1}{N} \sum_{i=1}^{k} maxj\,|ci \cap tj| \tag{2.13}$$

Where N = number of objects (data points), k = number of clusters, ci is a cluster in C, and tj is the classification which has the max count for cluster ci. When we say "correctly" that implies that each cluster ci has identified a group of objects as the same class that the ground truth has indicated.

2.8 Topic Modeling

Topic modeling is an unsupervised machine learning strategy that's able to identify a set of documents/texts, detecting word and phrase patterns within them. Also, group word collections that best characterize a set of reports. It is 'unsupervised' because topic modeling doesn't need any list of previously predefined tags or labeled by human beings for training.

2.8.1 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is one of the most frequent topic modeling methods analysts make use of. It is based on what is known as the distributional hypothesis which states that the semantics of words can be grasped by looking at the contexts the words appear in. In other words, under this hypothesis, the semantics of two words will be similar if they tend to occur in similar contexts.

That said, LSA computes how frequently possible words happen within the documents – and the total corpus – and expect that comparative archives will contain roughly the same conveyance of word frequencies for certain words. In this case, syntactic information (e.g. word order) and semantic information (e.g. the variety of implications of a given word) are ignored and each archive is treated as a bag of words.

2.8.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that document represented as random mixtures over latent topics, where each topic characterized by a distribution over words.

Latent Dirichlet Allocation (LDA) and LSA are based on the same implicit in assumptions: the distributional hypothesis, (i.e. similar topics make use of similar words) and the statistical mixture hypothesis (i.e. documents talk about several topics) for which a statistical distribution can be determined. The purpose of LDA is mapping each document in our corpus to a set of topics which covers a good deal of the words in the document.

The main difference between LSA and LDA is that LDA assumes that the distribution of topics in a document and the distribution of words in topics are Dirichlet distributions. However, LSA does not assume any distribution and thus, leads to more opaque vector representations of topics and documents.

There are two hyperparameters that control document and topic similarity, known as ***alpha*** and ***beta***, respectively. A low-value of alpha will assign fewer topics to each document whereas a high value of alpha will have the opposite effect. A low-value of beta will uses fewer words to model a topic whereas a high value will use more words, thus making topics more similar between them.

Topic models for texts viz., Latent Dirichlet Allocation, Dirichlet Compound Multinomial mixture, and von Mises–Fisher mixture model implemented in an online variant for text stream clustering by [25]. They concluded that vMF is better than the other two for cluster discovery. Further, a hybrid topic model proposed in [25] that uses both online and offline phases for efficient clustering.

2.9 Amharic Language

In this section, the literature's written on the Amharic language that is relevant for this thesis is briefly discussed. The historical details about the language, the alphabet, the punctuation, the numbers, and the Morphological complexity, and the syntactic discussed. In addition to these, the possible challenges that may occur while developing an automated system, which involves representing the Amharic language in the machine, indicated.

2.9.1 History about the language

Amharic (/aem'haerik/ or /a:m'ha:rik/ (Amharic: አማርኛ) Amarññа is one of the Ethiopian Semitic languages, which are a subgrouping within the Semitic branch of the Afroasiatic languages. It is spoken as a first language by the Amharas. Amharic is a Semitic language, related to Hebrew, Arabic, and Syriac. Next to Arabic, it is the second most spoken Semitic language with around 27 million speakers. Also, for a long period, it has been the principal literal language and medium of instruction and school subjects in primary and secondary schools of the country. Moreover, it is the working language of the Ethiopian Federal Government and some regional governments in Ethiopia, most documents in the country are produced in Amharic. There is an enormous production of electronic and online accessible Amharic documents.

According to [26], Amharic is one of the Semitic languages spoken in north-central Ethiopia. Next, to Arabic, it is the second most-spoken Semitic language in the world and it is the official working language of the Federal Democratic Republic of Ethiopia. It is also the native language of perhaps several million Ethiopian immigrants, especially in North America and Israel. It is the second-largest language in Ethiopia and possibly one of the five largest languages on the African continent. As a result, it has an official status and uses nationwide. Despite it has a large speaker population, the language has little computational linguistic resources.

2.9.2 Amharic Orthography

Unlike Arabic, Hebrew, and Syria (other Semitic languages), Amharic is written using a syllabic writing system, one originally developed for the extinct Ethiopian Semitic language Ge'ez and later extended for Amharic and other Ethiopian Semitic languages. As in other Abugida systems, each character of the Ge'ez (or Ethiopic) writing system gets its basic shape from the consonant of the syllable, and the vowel represented through more or less systematic modifications of these basic shapes.

The alphabet of the Amharic language consists of 33 core symbols or Fidel (ፊደል)Ėach of these core symbols occurs in seven different orders; the basic character plus six different symbols or orders formed from the basic character. There are also 37 further characters representing labialized variants of the consonants followed by particular vowels. The complete system has 268 characters. There is also a set of Ge'ez numerals.

2.9.3 Amharic Morph syntax

a) Amharic Morphology

Amharic is one of the most morphologically complex languages. Amharic nouns and adjectives are marked for any combination of number, definiteness, gender, and case. Moreover, they affixed with prepositions. For example, from the noun ተማሪ (tämari/student), the following words are generated through inflection and affixation: ተማሪዎች (tämariwoč/students), ተማሪው (tämariw/ the student masculine/his student), ተማሪየ (tämariyän/my student), ተማሪየን (tämariyän/my student objective case), ተማሪሽ (tämariš/your feminine student), ለተማሪ (lätämari/for student), ከተማሪ (kätämari/ from student), etc.

Amharic verb inflections and derivations are even more complex than those of nouns and adjectives consisting of a stem and up to four prefixes and four suffixes. The stem, in turn, is composed of a root, representing the purely lexical component of the verb, and a template, consisting of slots for the root segments and for the vowels (and sometimes consonants) that inserted around and between these segments. The template represents tense, aspect, mood, and one of a small set of derivational categories: passive-reflexive, transitive, causative, iterative, reciprocal, and causative reciprocal.

Amharic verbs are marked for any combination of person, gender, number, case, tense/aspect, and mood resulting in thousands of words from a single verbal root. As a result, a single word may represent a complete sentence constructed with subject, verb, and object.

2.9.4 Syntactic Structure of Amharic

Noun Phrases: An Amharic noun phrase has an explicit number, case, and definiteness. The accusative suffix appears obligatorily on definite direct objects and optionally on indefinite direct objects. An unusual feature of the language is the placement of the morphemes marking case (either the accusative suffix or one or another prepositional prefix) and definiteness [27] and [28]. These affixed to the noun itself only when it has no modifiers. If the noun has an adjective or relative clause modifier, the morphemes are normally affixed to the first of these." Headless noun phrases are common. These consist of one or more relative clauses and adjectives. Examples: tilliqun "the big one (acc.)", yägäzzawn "the one (acc.) that he bought".

Clauses: Unlike in other Semitic languages, all Amharic clauses headed by verbs [29]. The copula, näw (ነው) is a defective verb with only main clause present forms. Its past filled by the defective verb näbbär (ነበር) which also serves as the past of the defective verb of existence allä (አለ)İn other cases, the copula replaced by the perfect, imperfect, jussive-imperative, or gerund of either the verb norä "live" or the verb honä "become".

The basic word order of all Ethiopian Semitic languages is subject-object-verb (SOV), a feature that probably results from contact with Cushitic languages. As is common in SOV languages, the order of subject, object, and oblique arguments of the verb is somewhat flexible. In particular, for pragmatic reasons the subject can follow another argument: yohann1s mäskotun säbbäräw, mäskotun yohannis säbbäräw, "Yohannis broke the window". As in other Semitic languages, verbs agree with their subjects in person, number, and (in second and third person singular) gender. Verbs also agree with definite direct or indirect objects, but not both.

As in other Semitic languages, pronoun subjects and pronoun objects omitted unless they emphasized. This fact, in combination with the elaborate derivational and inflectional verb morphology, means that sentences consisting of a verb alone or the main verb and an auxiliary verb are uncommon: alt'äyyäqnatim "we didn't visit her", laflallaˇcihu "shall I boil (sth.) For you please?" awwaddädun "they made us like each other".

Either main clause verbs are in the perfect or a compound imperfect formed from the simple imperfect and conjugated suffix forms of the defective verb of existence allä. Subordinate clause verbs are in the perfect, simple imperfect, or gerund. tifäll1giyalläš "you (fem.sing.) Want", bitt1fälligi "if you (fem.sing.) Want".

Cleft constructions are very common in Amharic [29]. Indeed, for questions, cleft constructions are probably more common that non-cleft constructions. In a cleft sentence, the focused argument placed first, followed by the conjugated copula, followed by other arguments of the original verb, followed by the verb in the relative form: mindin näw yäsäbbäräw "what did he break?" lit. "What is it that he broke it"?

Relative clauses in Amharic consist of a relative verb and zero or more arguments and modifiers of the verb, as in any clause. A relative verb is a verb in either the imperfective or perfective with a prefix indicating relativism. As with the main clause verb, a relative verb must agree with its subject and may agree with its direct object if it has one. Both subjects and objects can relativized: yemiwedat sEt "the woman that he likes".

As noted above, when a noun is modified by a relative clause and has no preceding determiner, it is the relative clause that takes suffixes indicating definiteness or accusative case or prepositional prefixes: yetemereqew lj wendmE new," ' The boy who graduated is my brother." When a sequence of modifiers precedes a noun, it is the first one that takes the suffixes or prefixes: yetemereqew gwebez lj, 'the clever boy who graduated'.

Relative verbs agree with the main clause verbs that contain them. For example, yemiwedat alderesem," (He) who likes her didn't arrive", the third person singular masculine subject in the main clause verb agrees with the third person singular masculine subject of the relative clause verb.

Adverbial clauses usually indicated with prefix conjunctions on the relative form of the verb (in which case the initial yä is dropped) or the bare imperfect: silämmisäbräw "because he breaks it", bisäbräw "if he breaks it".

As is common in SOV (Subject-Object-Verb) languages [29], Amharic permits the chaining of a number of clauses together in a single sentence without explicit conjunctions indicating the relationship between the clauses. The usual interpretation is sequentially. All verbs but the final one appears in the gerund form. The final verb may be perfect, compound imperfect, jussive, or imperative. All of the gerund forms agree with the subject of the final verb. Example: bet tämälliso rat bälto täññа "He returned home, ate dinner, and went to bed" lit. "Returning home (3 pers.sing.masc.), eating (3 pers.sing.masc.) Dinner, he went to bed".

Chapter 3

Related Work

3.1 Introduction

Since the task of clustering is subjective, means that could be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the *similarity* among data points. There are many clustering algorithms known but few of the algorithms are used popularly. Text clustering algorithms are split into many different types such as agglomerative clustering algorithms, partitioning algorithms, and probabilistic clustering algorithms. This chapter presents different research work done in text clustering under different approaches.

3.2 Hierarchical clustering Approach

Hierarchical clustering algorithms received their name because they build a group of clusters that would describe as a hierarchy of clusters. The hierarchy can be in a top-down (called divisive) or bottom-up (called agglomerative) fashion. Hierarchical clustering algorithms are one of the Distanced-based clustering algorithms. It uses a similarity function to measure the closeness between a text document. The general overview of the hierarchical clustering algorithms for text data found in [30].

In the top-down approach, we begin with one cluster, which includes all the documents. We recursively split this cluster into sub-clusters. In the agglomerative method, each document is initially considered as an individual cluster. Then successively the most similar clusters merged until all documents embraced in one cluster. There are three different merging methods for agglomerative algorithms:

1. Single Linkage Clustering: In this technique, the similarity between two groups of documents is the highest similarity between any pair of documents from these groups.

2. Group-Average Linkage Clustering: In group-average clustering, the similarity between two clusters is the average similarity between pairs of doc-

uments in these groups.

3. Complete Linkage Clustering: In this method, the similarity between two clusters is the worst-case similarity between any pair of documents in these groups.

In [31] Hierarchical clustering algorithms have been employed to cluster documents with the help of instance and cluster level constraints. Their work has been considered must-link and cannot-link constraints. They believe the use of such constraints in hierarchal clustering is the best way to find specific kinds of the cluster and avoid others. They test their state of the art with six real-world UCI datasets. They find that the cluster-level constraint can reduce the computational time between two and four-fold by effectively creating a pruned dendrogram (A tree diagram used to show the arrangement of data into clusters). To further improve the efficiency of agglomerative clustering they have been introduced the constraint that allows the use of the triangle inequality to save computation time.

The work named *"Evaluation of Hierarchical Clustering Algorithms for Document Datasets"* [32] hierarchal clustering has been evaluated for document dataset and perform comparison analysis with partitioned algorithms. Additionally, they had a present new class of clustering algorithm named constrained agglomerative algorithms. This algorithm combines the features of both partitioned and agglomerative algorithms. In their experiment, they have been used twelve datasets with the smallest and largest dataset contains 878 and 4,069 documents respectively.

3.3 Probability Clustering and Topic Models Approach

Topic modeling is one of the most popular probabilistic clustering algorithms which has gained increasing attention recently. The main idea of topic modeling [33] is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics while a topic is probability distribution over words.

The two main topic models are *Probabilistic Latent Semantic Analysis (pLSA)*

[34] and *Latent Dirichlet Allocation (LDA)* [33]. [34] Introduced pLSA for document modeling. pLSA model does not provide any probabilistic model at the document level, which makes it difficult to generalize it to model new unseen documents. [33] Extended this model by introducing a Dirichlet prior to mixture weights of topics per documents and called the model Latent Dirichlet Allocation (LDA).

In [35] the authors have been employed two different frameworks for unsupervised topic modelling of CompWHoB Corpus, a political corpus collecting the transcripts of the White House Press Briefings. To achieve their goal first, they had employed LDA model approach by extracting from each answer/question document only the topic with the highest probability. Secondly, they had applied the word embedding is generated from the Word2Vec model [36] to their data to test how dense high-quality vectors represent our data. Finally, they have been compared the result to show which one performs a better result on the given dataset and Results show that the use of word embeddings outperforms the LDA approach but only if a linguistic task-oriented preprocessing stage is carried out with purity 0.54 and 0.46 respectively.

The work in [37] have been proposed a generative model, which integrates document clustering and topic modeling. Given a corpus, they have been assumed there exist several latent groups and each document belongs to one latent group. Each group possesses a set of local topics that capture the specific semantics of documents in this group and a Dirichlet prior expressing preferences over local topics. Besides, they have been assumed there exist a set of global topics shared by all groups to capture the common semantics of the whole collection and a common Dirichlet prior governing the sampling of proportion vectors over global topics for all documents.

The accuracy of topic modeling-based clustering methods including LDA + K-Means, LDA + Naïve are generally better than K-means, normalized cut, and factorization-based methods. This corroborates their assumption that topic modeling can promote document clustering. The semantics discovered by topic models can effectively facilitate accurate similarity measure, which is helpful to obtain coherent clusters.

Latent Dirichlet Allocation (LDA) model

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents represented as random mixtures over latent topics, where each topic characterized by a distribution over words.

LDA assumes the following generative process for each document w in a corpus D:

1. Choose $N \sim$ Poisson (ξ).
2. Choose $\theta \sim$ Dir (α).
3. For each of the N words wn:
    - Choose a topic $zn \sim$ Multinomial (θ).
    - Choose a word wn from $p\ (wn\ |zn,\beta)$, a multinomial probability conditioned on the topic zn.

More precisely a document of N words w $=\langle w1,\ w2,\ldots\ldots\ldots\ldots wN\rangle$ is generate by the following process:

First θ is sampled from a Dirichlet $(\theta 1,\ ..\ .....\theta k)$ distribution. This means that θ lies in the $(k-1)$-dimensional simplex: $\theta i \geq 0, \sum_i \theta i = 1$. Then for each of N words at topic $Zn \in \{1,\ldots\ldots.k\}$ is sampled from $Mult\ (\theta)$ distribution $p\,(Zn = i \mid \theta) = \theta i$. Finally, each word wn is sampled, conditioned on the Znth topic from multinomial distribution $p(w|Zn)$. Institutively, θi is can be thought of the degree to which topic i is referred to in the document. The probability of a documents is therefore:

$$p\,(w) = \int_\theta \left(\prod_{n=1}^{N} \sum_{Zn=1}^{k} p(wn|zn;\beta)p(Zn|\theta) \right) p(\theta;\alpha)d\theta$$

Where $p(\theta;\alpha)$ is Dirichlet, $p(w|Zn)$ is multinomial parametrized by θ. And $p(wn|zn;\beta)$ is multinomial over words. The model is parametrized by k-dimensional Dirichlet parameters $\alpha = \langle \alpha 1,\ldots\ldots.\alpha k\rangle$ and a $K \times |V|$ matrix β which are parameters controlling the k dimensional distribution over words.

The general architecture of LDA model depicted in figure 3.1 below.

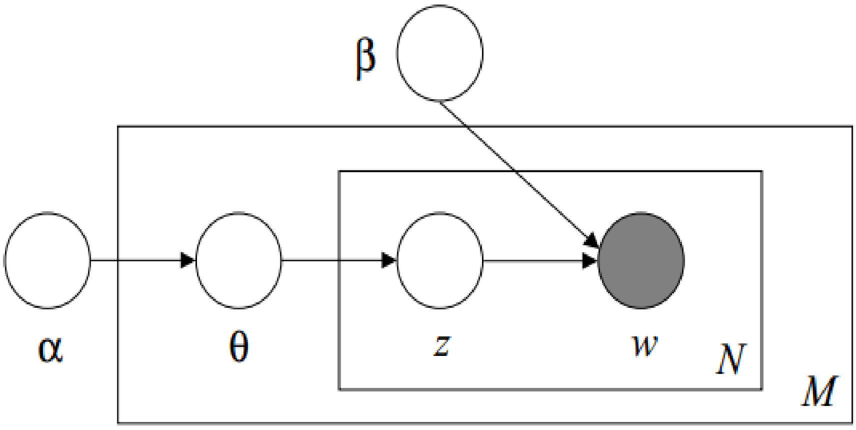

Figure 3.1: Graphical model representation of LDA [33].

where:

M - denotes number of documents

N - is the number of words in a given documents (document I has Ni words)

α - is the parameter of the Dirichlet prior on the per-document topic distributions

β - is the parameter of the Dirichlet prior on the per-topic word distribution

θ - is the topic distribution for document

z - is the topic word in document

w - is the specific word.

Figure 3.1 represents graphical model of LDA as probabilistic model. As the figure makes clear, there are three levels to the LDA representation [33]. The parameters α and β are corpus level parameters, assumed to be sampled once in the process of generating a corpus. The variables θd are document-level variables, sampled once per document. Finally, the variables zdn and wdn are word-level variables and sampled once for each word in each document.

3.4 Partition Clustering Approach

Partitioned clustering algorithms compute a k-way clustering of a set of documents either directly or via a sequence of repeated bisections. A direct k-way clustering commonly computed as follows. Initially, a set of k documents selected from the collection to act as the seeds of the k clusters. Then, for each document, its similarity to this k seeds computed, and it was assigned to the cluster corresponding to its most similar seed. This forms the initial k-way clustering. This clustering is then repeatedly refined so that it optimizes the desired clustering criterion function. A k-way partitioning via repeated bisections obtained by recursively applying the above algorithm to compute 2-way clustering (i.e., Bisections). Initially, the documents partitioned into two clusters, and then one of these clusters selected and further bisected, and so on. This process continues k -1 times, leading to k clusters. Each of these bisections performed so that the resulting two-way clustering solution optimizes the particular criterion function.

The author in [15] the overall k-way clustering solution will not necessarily be at local minima with respect to the criterion function. The key step in this algorithm is the method used to select which cluster to bisect next. In all of our experiments, we chose to select the largest cluster, as this approach lead to reasonably good and balanced clustering solutions [15]. Extensive experiments presented in [38], show that the clustering solutions obtained via repeated bisections are comparable or better than those produced via direct clustering are. Furthermore, their computational requirements are much smaller, as they have to solve a simple optimization problem at each step. For this reason, in all of our experiments we use this approach to compute partition-clustering solutions.

K-Means

K-means is the most widely used clustering technique; it belongs to the class of iterative centroid-based divisive algorithm. The algorithm tries to determine k partitions that minimize the squared-error function. The k-means method can apply only when the mean of cluster defined. The k-means algorithm for partitioning based on each cluster's center, which represented by the mean value of the objects in the cluster [14].

The author in [39] tries to combine the largest minimum distance algorithm and traditional K-Means to improve the document clustering. This improved algorithm can make up the shortcomings of the traditional K-Means algorithm to determine the initial focal point. The improved K-Means algorithm effectively solved two disadvantages of the traditional algorithm, the first one is greater dependence to choose the initial focal point, and another one is easy to trap at a local minimum. Testing the efficiency of the improved K-Means algorithm composed of 20 random data and classified into five classes according to the degree of the cluster. According to the academic analysis and result of the experiment, the improved K-Means not only keeps the high efficiency of standard K-Means but also raises the speed of convergence effectively by improving the way of selecting the initial cluster focal point. The improved K-Means is obviously better than standard K-Means in both cluster precision and stability.

3.5 Text Categorizer for Amharic Language

As stated in [13], Zelalem had investigated the characteristics of Amharic news items for the Ethiopian News Agency (ENA) and designed a prototype that has the capability of automatically classifying news items into their predefined classes based on their content. Zelalem had applied statistical techniques of automatic classification. Statistical techniques include document analysis, generation of document and class vectors based on document and class representatives, and matching document and class vectors to determine the class of a document. The system can classify new documents by matching the document vector with the centroid vector of each class. The document is routed to the most similar class. The similarity between the document vector and the class vectors computed to determine the class in which the document belongs.

The other research work in this domain is the one done by (Abraham, 2013). Their work mainly focused on applying item sets method to categorize Amharic documents. In addition to that, the implementation of all the required tools, which helps to carry out automatic Amharic Document categorization using item sets method, was developed and the algorithm examined. According to this proposed work experiment results item sets method is an efficient method to categorize Amharic documents.

In (Abraham, 2013). the author has been investigated local document categorizing. The author had addressed the application of machine learning techniques to automatic document categorization of Amharic news items. The machine learning techniques NB and K-Nearest Neighbors (KNN) classifier used. As the author stated, the main requirement of the classification scheme is to provide sufficient background information on any topic. The tool supports different classification methods such as NB, KNN, TFIDF, SVM, and probabilistic. The average precision, recall and F1 values obtained are 96%, 97% and 96% respectively.

Mulualem Wordofa [12] did Amharic document clustering using semantic indexing for information retrieval. This work depended on the term frequencies. In this work, a document summary for each cluster containing the unmistakable terms whose frequencies are high is set up in the wake of preprocessing of the document. The author utilized K-means partitioning. As the experiments have shown semantic indexing has improved the performance of Amharic information retrieval system from 60% to 66% F-measure.

3.6 Text Clustering for Non-Amharic Languages

In [40] the author uses two major topic model approaches. Namely using basic topic model; and based on cluster-oriented topic model. At the end the author evaluated the performance of the two approaches. The experimental result shows that simple method can achieve better clustering accuracy and recall than cluster-oriented.

Another research work with topic model to cluster scientific documents is done in [41]. The author considers grouping of several collection of academic papers into several cluster based on their content using topic modeling. To evaluate the proposed method the author, collect a number of academic research papers from seven different fields. As the result shows topic model results better than four topic modeling algorithms.

The study in [42] try to cluster chinse corpus into sentence level using k-means algorithm and a continuous vector representation of sentences approach. According to the paper sentence, clustering is appealing problem in text clustering in which a document may made up of only one single sentence. This problem has been receiving special attention in the natural language processing (NLP) community since it allows for training specific models for each of the obtained clusters, leading to more task-focused models. Sentence clustering can also be of interest in other NLP tasks, such as done for text recognition or statistical machine translation.

The authors in [43] introduced a novel clustering model based on the combination of Latent Dirichlet Allocation (LDA) and Word2Vec skip-gram model. The model refines the information of short texts from academic abstracts according to the feature of paragraphs and it generates topic embedding's' containing more information compared with BOW model. It uses less data to train the word embedding and probability matrix. They have shown that this method has better performance than some traditional ones.

The work in [44] tries to cluster Arabic text with improved clustering algorithm and dimensionality reduction. This research proposes three approaches; Unsupervised, Semi-Supervised techniques, and Semi-Supervised with dimensionality reduction to construct a clustering-based classifier for Arabic text documents. After document, preprocessing removing stop words and gets the root for each term in each document. They apply a term weighting method to get the weight of each term to its document. Then apply a similarity measure method to each document and its similarity with other documents. Also, using F-measure, entropy, and support vector machine (SVM) to calculate accuracy.

The authors in [45] have present an approach to cluster from small to medium texts corpora containing very short texts based on semantic enrichment of texts. The semantic enrichment in the preprocessing step is a general-purpose approach. It expands the initial texts with additional features (tokens representing categories, synsets, glosses, hypernyms, or similar words), and does not influence the text clustering methods. The clustering methods still receive texts as input, which can interpret as a bag of words, so this methodology can be very easily deployed. The experiments concerning semantic enrichment for the data sets showed that among the approaches using BabelNet tools, only the Babel-synsets approach provided better results, whereas the other approaches did not improve the clustering quality.

Previous works for local texts categorizing mainly focused on counting the frequency of the appearance of a word in a given document. The main problem of this approach is that it does not consider the semantic relation between words. What's more that, these approaches not feasible for short texts because the texts have only a few hundred words. The work of this thesis proposes the use of a topic modeling method to cluster short texts. Therefore this thesis work attempt to overcome the limitation of the above work with topic modeling specifically LDA (Latent Dirichlet Allocation) to cluster short Amharic texts. Moreover, it uses neural word embedding (Skip-gram model) for feature extraction to improve the clustering result which has not been used in the previous works.

Author/s	Approach used	Description
Mulualem Wordofa	Document clustering using semantic indexing for information retrieval (Bag of words approach)	The author had tried to categorize Amharic news item. Use frequency of appearance of a word to construct feature space of the text. The limitation of the approach is that it is not feasible for short text due to short texts has few words only.
Meron Sahlemariam	Use news ontology	The author has used news ontology to categorize local news item based on concept. But the limitation of this approach is that it is very difficult to construct an ontology which can cover all the news concept especially when news come from different sources.
Zelalem Sintayehu	Applied statistical techniques ofautomatic classification	The author had applied statistical method to investigateclassification of Amharic news item based on content. The proposed approach considers presence or absence of words to construct vector which is not appropriate for short texts.

Chapter 4

Design and Implementation of Short Amharic Text Clustering using Topic Modeling

4.1 Introduction

This chapter briefly describes the proposed design and implementation of short Amharic text clustering using a topic modeling approach. In the design and implementation process of short Amharic text clustering, the main activities include preprocessing, topic modeling, and clustering. Preprocessing activities include tokenization, normalization, stemming, and stop word removal.

Topic modeling has some steps that make the texts suitable for topic extraction. This module identifies latent/hidden topics in short text-using LDA as described in section 3.3 in detail and cluster texts into a different group based on the feature extracted (topics). Moreover, all activities related to the implementation presented in this chapter.

4.2 Building Corpus

We had collected and built a corpus by crawls different news agencies' websites and from a set of publicly available short Amharic text collections. Mainly based on news reports in local Amharic newspapers. The data source for this study collected from `https://www.fanabc.com/`, `http://www.waltainfo.com/`, `http://www.zhabesha.com/` and `https://www.ena.et/am/`. The collected text has multiple categories, such as health, art, politics, science and technology, sport, and others.

4.3 Architecture of Short Text Clustering

As mention in chapter 1, the main aim of this thesis work is to use topic modeling as a way of improving short text clustering. Figure 4.1 below shows the general architecture of short Amharic text clustering using topic modeling. It is structured into two modules based on the data and process flow between the components. The preprocessing module is responsible for the target text processing. The topic modeling module is responsible for using LDA for identifying latent or hidden topics from short texts. The input for the system is a short text and the output will be a set of clustered texts under different topics.

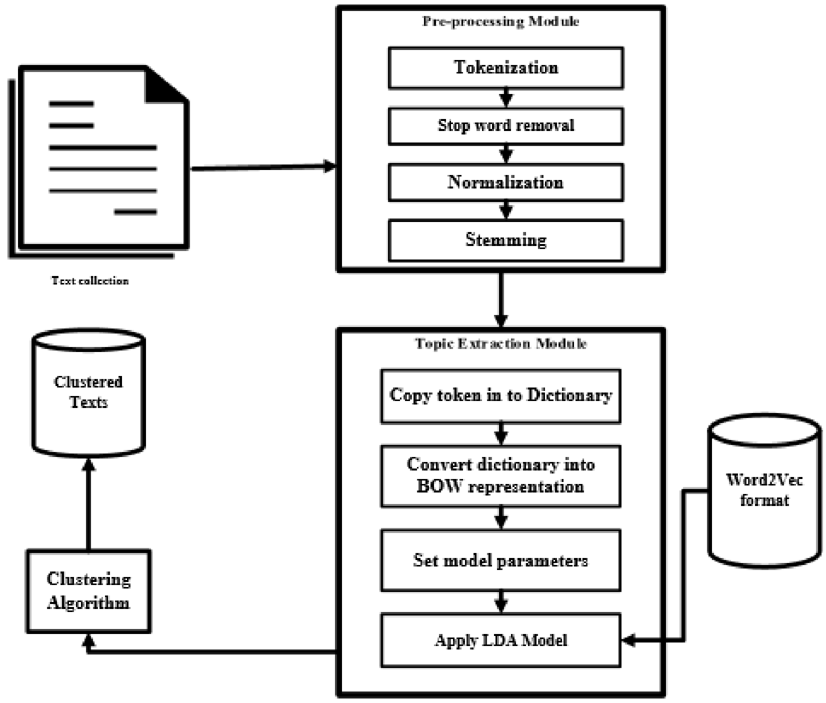

Figure 4.1: General architecture of short Amharic text clustering using topic modeling

As shown in 4.1 the input of the system is a set of short texts. After preprocessing the input texts, the preprocessing module will generate a set of preprocessed text. The topic modeling module attempts to use LDA to find hidden (latent) topics from the preprocessed text. After that, it will divide the text into different categories based on the topic.

4.3.1 Pre-processing Module

As we have discussed at the beginning, the goal of this module is to generate a set of preprocessed texts, which are important for clustering activities. As we reviewed in Chapter 2, the existing clustering methods currently rely on the frequency of words in the text and the inverse frequency of the document. However, these approaches have limitations not to use in this thesis work. First short texts have no enough word to consider and those texts are very sparse and noisy so that they cannot be clustered using the above methods. Additionally, those methods are not able to consider the latent topics in the texts.

Instead, we rely on topic modeling that can address the above limitations. An approach automatically finds hidden (latent) topics from the given corpus (in our case short Amharic text corpus). To do so, we need to first preprocess the Amharic text corpus that increases performance and reduces the runtime of the clustering. This process is language-dependent and includes the following activities: - tokenization, removing unnecessary words, changing characters and words into their common form, sub-sampling frequent words, and stemming.

A. Tokenization

Tokenization is the process of splitting up the given text into units named tokens or it describes splitting text sentences into individual words. This is done by locating word boundaries between two individual words. The tokens may be words, number, punctuation marks, special symbols, etc. In Amharic a common way to split a text is using whitespace. It also considers compound words like ቤት- መኪና as hyphenated to keep the meaning of the word.

B. Removal of Extraneous Characters

Algorithm 1 Algorithm to remove Extraneous Characters

```
Store special characters in temporary variable
Read file name
for each file name in corpus do
    Split into word and store in list
    for Each element of the list do
        if An element is one of the special character then
            Discard the element
        else
            Return element.
    Endfor
Endfor
```

The numbers, dates, punctuation marks and control characters in the text of each file not considered for building a topic model, as they do not provide important information about target word meaning. Words containing numbers like (2^{nd} i.e. 2 ^ኛ} or ኢዜአ03862) were excluded at the first phases of preprocessing. Moreover, the standard control character; Amharic punctuation marks ፡ ፣ ፤ ። ፥ ፦ and symbols borrowed from other languages (?,! , “,”, ‘, /, \, etc.) were ignored.

C. Normalization

In Amharic, there are words, which can spell in different formats. It would unnecessarily increase the number of words considered during topic modeling that could reduce the efficiency and accuracy of the clustering. Hence, this activity normalizes these spelling variations by changing the different forms of a character into one common format. The other normalization issue related to the shorthand representation of words like አ/ ኣ, ት/ ሚ, and ጽ/ ቤት Hence, these forms have been converted into their expanded long forms. Table 4.1 shows an example of the character redundancy where more than one symbol is used for the same sound.

Consonants	Other symbols with the same sound
ሀ (hä)	ሃ ሐ ሓ ኃ ኀ ኻ
ሰ (sä)	ሠ
አ (ä)	ኣ ዐ ዓ
ጸ (tsa)	ፀ

Table 4.1: Redundant Amharic characters

The algorithm that perform character normalization is given below.

Algorithm 2 Algorithm to Normalize Character

Read token of words as input

Find character with double values

Replace with common character

Return normalized character/s

D. Stemming

The other normalization activity is stemming the process of reducing inflected (or sometimes derived) words to their stem. In this work, it is sufficient that related words map to the same root(even if this stem is not in itself a valid root), and it provides a means to reduce index terms and hence save storage space, maximizes the performance of cluster (accuracy and efficiency). In this thesis work, we employ the stemming algorithm developed in [46]. Normally proper names, dates, and numbers (i.e. resources and values) not to be subjected to stemming since they will not be reduced to root words.

E. Stop-word Removal

It removes the most frequently occurring words from the text that do not provide important information about the meaning of the target words. The assumption is that words, which occur frequently in almost all text, are non-informative. Removal of frequent words during training results in a significant speedup (around 2x - 10x), and improves the accuracy of the representations of less frequent words [47]. Like other languages, some words in Amharic are used very frequently in the normal usage of the language.

4.3.2 Topic Modeling Module

The topic modeling module includes activities like copy the tokens into dictionary format; convert the dictionary into BOW representation.

Once the preprocessing module performs its task it will give the preprocessed text to the topic modeling module. The topic-modeling module will try to find the 'topics' from the preprocessed text.

A topic model is a type of statistical machine learning model for discovering the abstract "themes" that occur in a document set. Topic modeling is a widely used text-mining method for a finding of hidden semantic structures in a body of the text. Intuitively, given that a document is about a specific subject, one would expect similar terms to appear in the document more or less frequently: 'ጤና' (t'ēna/Health), 'ህክምና' (hikimina/Treatment), 'በሽታ' (beshita/Disease), 'ሆስፒታል' (hosipītali/Hospital), 'ጤና_ጣቢያ' (t'ēna_t'abīya/ Health Station) , 'ህብረተሰብ' (hibiretesebi/Society), 'ደም' (demi/Blood) , 'ጤና_ጥበቃ' (t'ēna_t'ibek'a/ Health Protection) 'ወባ' (weba/Malaria), 'ህመም' (himemi/Pain), 'ክትባት' (kitibati/Vaccination), 'መድሀኒት' (medihānīti/Medicine), and 'ምርመራ' (mirimera/Investigation) will appear more frequently in texts about ***Health***.

In texts about ***Sport*** 'ዋንጫ'(wanich'a/Cup), 'ቡድን' (budini/Team), 'ኳስ' (kwasi/Ball),

‘እግር_ኳስ’ (igiri_kwasi/ Football), ‘አሰልጣኝ’(āselit’anyi/Coach), ‘ሊግ’ (līgi/League), ‘ጨዋታ’(ch’ewata/Game), ‘ሩጫ’ (ruch’a/Run), ‘ፕሪሚየር_ሊግ’ (pirīmīyeri_līgi/ Premier League), ‘ጎል’ (goli/Goal), ‘አሸነፈ’(āshenefe/Won), ‘ፊፋ’ (fīfa/FIFA), ‘አትሌቲክስ’ (ātilētīkisi/Athletics), ‘ተጨዋች’ (tech’ewachi/Player), ‘ግጥሚያ’ (git’imīya/Match), ‘ደጋፊ’ (degafī/Fan), ‘ዳኛ’ (danya/Judge), ‘አትሌት’ (ātilēti/Athlete), and ‘ስታዲየም’ (sitadīyemi/Stadium) will appear. As usual, stop word and frequent words will appear in both texts.

Topic modeling addresses the following type of problem: you have a collection of documents (emails, survey responses, service tickets, product reviews, etc.), and you want to find out the various topics that they cover and group them by those topics.

The way these algorithms operate is by suggesting that each text is consisting of a mixture of topics. And then trying to figure out how strongly each topic has a presence in a specific document. This is achieved by grouping together the texts based on the terms they contain, and finding similarities between them.

LDA is a type of unsupervised machine learning topic model which scanning a set of documents (referred to in the NLP field as a corpus), examines how words and expressions co-emerge in them, and consequently “learns” groups or groups of words that best describe those documents. These arrangements of words regularly seem to speak to a reasonable subject or theme.

In somehow, every topic-modeling algorithm starts with the presumption that your documents consist of a fixed number of topics. The model then assesses the evaluate the basic structure of words in your datum and endeavors to find the groups of words that best “fit” your corpus-based on that constraint.

For the proposed work, we will use LDA (latent Dirichlet allocation). In LDA, each document is viewed as a mixture of various topics where each document considered having a set of topics assigned to it via LDA. This is identical to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution assumed to have a sparse Dirichlet prior. The inadequate Dirichlet priors encode the instinct that documents cover only a small set of topics and that topics use only a small set of words that appear frequently. In fact, this result leads to a high degree of disambiguation of words, and the task document of the topic becomes more and more accurate.

Imagine a fixed arrangement of topic. We characterize every topic as spoke to by an (unknown) arrangement of words, those topics are that our writings spread, however we don't have the foggiest idea what they are yet. LDA attempts to outline the (known) records to the (unknown) subjects in a way with the end goal that the words in each document generally caught by those topics. Reports with a similar subject will utilize comparative words. It accepted also that a blend of topics makes each document, and each word has a likelihood of having a place with a specific topic.

LDA expect documents are created the accompanying way: pick a blend of topics (state, 20% subject A, 80%, subject B, and 0% subject C) and afterward pick words that have a place with those subjects. The words are picked aimlessly as indicated by the fact that they are so prone to show up in a specific document as appeared in the figure 4.2 below.

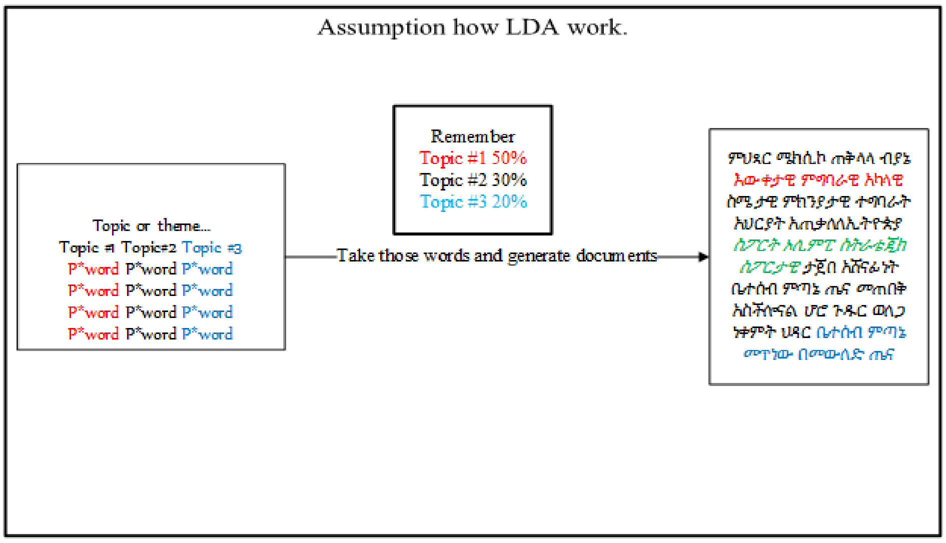

Figure 4.2: Assumption how LDA works

Obviously, in actuality, documents are not composed along these lines that would be mad. Documents composed by people have qualities that make them meaningful, for example, word requests, sentence structure, and so forth. However, it can contend that just by taking a gander at the expressions of a document, you can identify the topic, regardless of whether the real message of the document doesn't come through.

This event is the thing that LDA does. It saw a report and accepted it to be generated as described earlier. At that point, it works in reverse from the words that make up the document and attempts to figure the blend of subjects that brought about that specific game plan of words. See the figure 4.3 below.

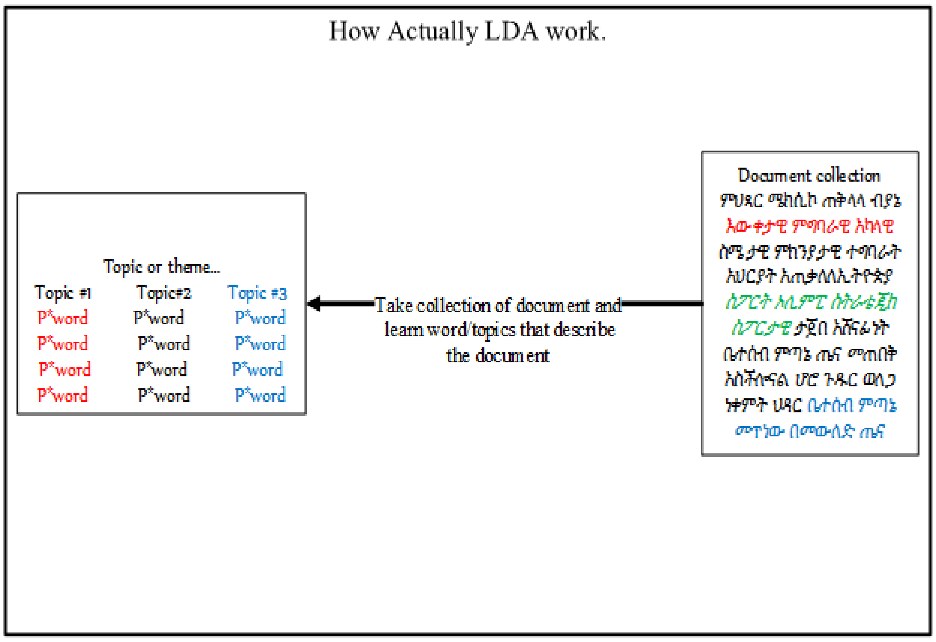

Figure 4.3: How LDA actually works.

Something we should specify about the execution is that it has two hyper parameters for preparing, ordinarily called α (alpha) and β (beta). Recognizing what these do is significant for utilizing libraries that actualize the implementation.

Alpha controls the similitude of documents. A low value will speak to reports as a mixture of barely any subjects, while a high value will yield documents characterizes of more topics - making all the documents show up increasingly like one another.

Beta is the equivalent but for topics, so it controls topic closeness. A low value will represent topics as increasingly unmistakable by making less, interesting words have a place with every topic. A high value will have the contrary impact, bringing about topics containing more words in like manner.

To support the above idea let us give our sample-trained model with some of the selected datasets shown in the image below.

We have set our training model algorithm as follow:

Algorithm 3 LDA model training Algorithm

Input: Short Amharic Texts

Output: Trained LDA model

Start:
Read Datasets
Pre-process the data-sets
Split the data into word and store into list
Copy the list into Dictionary
Convert dictionary into bag of words representation
Set appropriate LDA model parameters
Train the model
Save the Model
Stop.

Apart from that, alpha and eta are hyperparameters that affect sparsity of the topics. According to [49] in Gensim both defaults to 1.0/num_topics prior. Chunksize is the number of documents to be used in each training chunk. update_every determines how often the model parameters should be updated and passes is the total number of training passes.

4.3.3 Neural Word Embedding (Word2Vec)

This subcomponent focuses on constructing the semantic representation of words based on the statistical distribution of the co-occurrence of words in the text corpus. The meaning of a word is represented by its corresponding vector. We use the Skip-gram Word2vec algorithm to build a semantic model, which includes the semantic representation of all words in the preprocessed Amharic corpus. Skip-gram is a predictive neural word embedding algorithm. It uses the current word to predict the surrounding window of the context word. Compared with the earlier distribution model algorithm, it has many advantages. [36].

Skip-Gram Model

The objective of the model is to find word representations that are useful for predicting the co-occurring words in a given texts. According to [36] to predict c context words having one target word on the input as it is shown in the figure 4.4 below. More formally, given a sequence of training words w_1, w2, w3, . . ., wt., the objective of the model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0}\text{logp}\left(w_{t+j}\mid w_t\right) \tag{4.1}$$

Where -c and c are limits of our context window (size of context window). The basic model formulation defines $p(w_{t+j}\ |w_t)$ using the softmax function:

$$p\left(wO|wI\right)=\frac{exp\left(v'_{wO}{}^{T}v_{wI}\right)}{\sum_{w=1}^{W}exp\left(v'_{w}{}^{T}v_{wI}\right)} \tag{4.2}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the total number of words in preprocessed Amharic corpus.

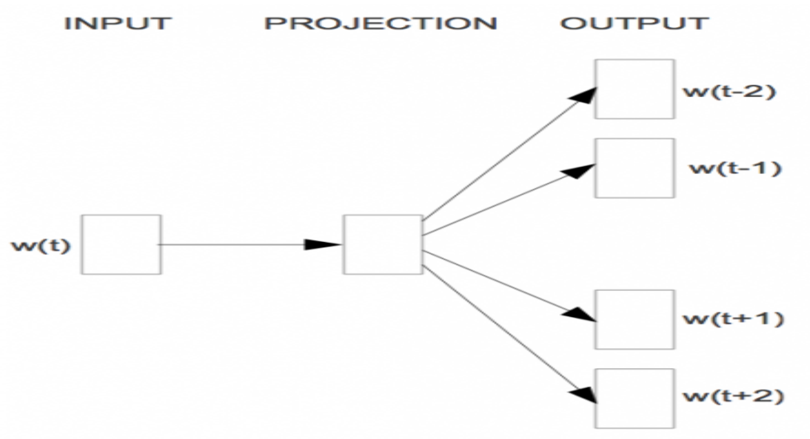

Figure 4.4: Skip gram model.

We have implemented our proposed model using LDA as its fragment code shown below. The LDA model has some parameters. the parameters include corpus, update_every, chuncksize, alpha, passes, and etc.

```
from gensim import corpora,models
id2word=corpora.Dictionary(text)
corpus=[id2word.doc2bow(text) for text in text]
num_topic=6
```

```
lda_model=models.LdaModel(corpus, num_topics=num_topic,\
    id2word=id2word,update_every =0, \
    chunksize=300, passes=30,\
    alpha=0.5 per_word_topics=True)
```

4.3.4 Clustering Algorithm

Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

Spherical K-means

Spherical k-implies is the most well-known strategy for clustering text data where the calculation takes cosine similarity between information [23]. In the clustering process, each cluster means vector refresh, just after all report vectors have been appointed, as the (standardized) normal of all the text vectors appointed to that group. The spherical k-means calculation looks like:-

- Normalize each data point.
- Clustering by finding center with minimum cosine angle to cluster points.
- Similar iterative algorithm to basic k-means.

Finally, clustering activity which is assigning unseen text to a certain cluster with a clustering algorithm. The algorithm that perform the clustering is given as:

Algorithm 4 Algorithm to cluster text

```
Load saved cluster features
Read test set texts
Pre-process
for Each file name in test set do
    Calculate similarity with the cluster features
    if text is the similar to one of the clusters then
        Cluster text to the most similar cluster feature
        Return cluster ID
    else
        Return text not clustered
    EndIf
Endfor
```

Chapter 5

Experiment and Evaluation

5.1 Introduction

In this chapter, we have present an evaluation of the proposed framework. The evaluation of the framework is an important part of any work to check whether it meets its goal or not. To conduct the experiment, we have followed a set of procedures, which consist of a set of activities. In the subsequent pages of this thesis, we will discuss the procedures and the results.

5.2 Experimental Procedure

To assure how our thesis work meets the design goals we conducted experiments for each class of text. In the following Section the evaluation metrics along with the corresponding results of the proposed short Amharic text clustering system, data collection, sample selection techniques will be presented. The standard methods that are used to evaluate a clustering activity are used to evaluate the performance of the system. We follow necessary procedures for experimental activities to conduct the experiment from the beginning to the end.

Before we start performing actual clustering it is important to identify the appropriate number of clusters (K) for our data-set. We have used a graphical tool named Average Silhouette Method.

Average Silhouette Method:

Briefly, it quantifies the nature of a clustering. That is, it decides how well each item exists in its cluster. A high average silhouette width demonstrates a good clustering.

Average silhouette technique figures the average silhouette of perceptions for various estimations of k. The ideal number of cluster k is the one that maximize the average silhouette over a range of potential qualities for k [50].

The algorithm is similar to the elbow method and can computed as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different val-

ues of k. For instance, by varying k from one to 10 clusters.

2. For each k, calculate the average silhouette of observations (*avg.sil*).

3. Plot the curve of *avg.sil* according to the number of clusters k.

4. The location of the maximum considered as the appropriate number of clusters.

As we see in the figure 5.1 below the appropriate number of cluster *K* for our dataset is six.

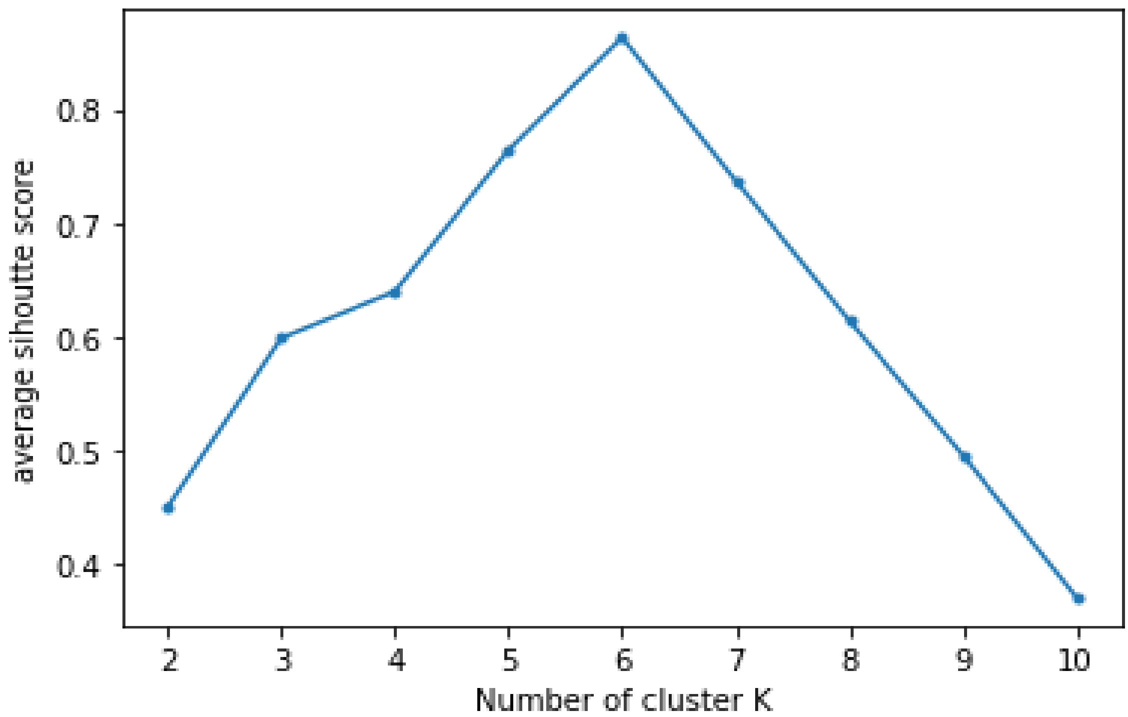

Figure 5.1: Silhouette score to identify number of cluster K.

Depending on the above experimental result the appropriate number of cluster (K) for our dataset has been set to six.

5.2.1 Data Collection

As mentioned in 4.2, we have considered Amharic short news texts to develop the corpus for the work of this thesis and demonstrated the process of automatic clustering of Amharic short texts. The collected news items are only news items with only a few hundred words. The collected texts have a number of categories such as Health, Art, Politics, Science and Technology, Sport, and Others as they are categorized in the news website. Total number of texts collected for this work is Art=1016, Health=446, others=1040, Politics=1013, Sport=877 and Science and Technology=1308, total 5700 short Amharic texts.

5.2.2 Sample Selection

Of the 5700 short Amharic news items collected from different local news sites, 5174 texts have been used for training, and 526 of them are prepared for testing. Which means for each and every text class we have used 90% of the data for training and the rest 10% for testing. We have used probability sampling techniques which is appropriate for unsupervised machine learning. The data preparation includes preprocessing the texts. The test set texts come from the "art", "health", "sports", "politics", "other" and "science and technology" text categories.

5.2.3 Prototype Development For Training

A prototype is an early sample, model, or release of a product built to test a concept or process. It is a term used in a variety of contexts, including semantics, design, electronics, software programming, and machine learning. Prototypes can help evaluate new designs to improve the accuracy of system analysts and users. Prototype design aims to provide specifications for the actual working system, not for the theoretical system. In some design workflow models, creating a prototype is the step between the model development and the evaluation of an idea.

The prototype used for training the model is depicted in figure 5.2 below. The process of training a model involves providing a machine learning algorithm (that is, the learning algorithm) with training data to learn from. The machine learning model refers to the model artifact that is created by the training process. As it is shown in the figure, the input for the system is a short Amharic text. The input texts are preprocessed using the preprocessing module before the LDA method has been employed to extract the topic. LDA topic extraction is followed by feature extraction and then clustering. The prototype has a number of components like input corpus (short Amharic text), preprocessing module, LDA method, feature extraction, and finally, clustering based on features extracted. A detailed description of each component is given.

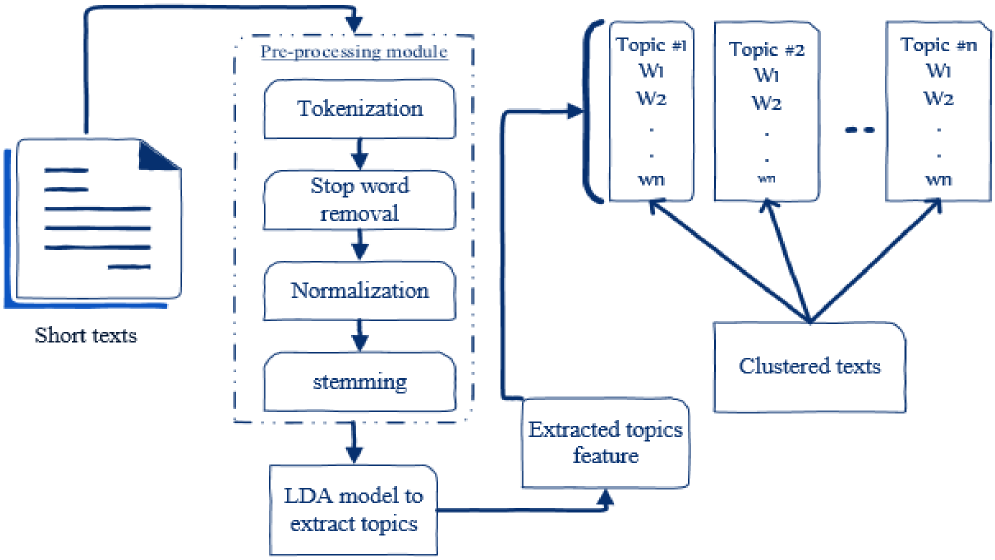

Figure 5.2: Training Prototype.

The first component for the above figure is a collection of short Amharic text collected from different local news agencies like Fanabc.com, waltainfo.com, ena.com, etc. Form the collected texts training datasets were used to train the model. After the training datasets get into the system preprocessing module has been applied to them to process for better training.

The first component of preprocessing module is tokenization. It is splitting sentence into tokens. The tokens may be words, number, punctuation marks, special symbols, etc.

Another sub-component of the preprocessing module named normalization was used to normalize different Amharic characters into a single common format. In Amharic, there are words, which can write in different formats. It would unnecessarily increase the number of words considered during topic modeling which could reduce the performance of the clustering activity. Therefore, this component normalizes these spelling variations by changing the different formats of a character into one common form. For example, the characters ሃ, ሐ, ሓ, ኃ, ኀ, ኻ, ሀ should be normalized into common format ሀ (**hä**) and the same is true for another Amharic redundant characters. Another normalization issue is related to shorthand representation of words like አ/ አ, ት/ ሚ

and ጽ/ ቤት. Therefore, these formats should convert into their expanded long formats.

The next component of the preprocessing module is stemming. As usual, stemming was used to change inflected/derived words into their root form. In this thesis work, it is sufficient that related words map to the same stem (even if this stem is not in itself a valid root). For this thesis work proper names, dates, and numbers (i.e. resources and values) not to be subjected to stemming since they will not be reduced to root words.

Preprocessing module complete its task by removing stop words (words that appear most commonly in all texts but not necessary for topic modeling). It also remove extra unnecessary character likes punctuation, numbers dates.

The preprocessed text has been pass though LDA (Latent Dirichlet Allocation) method so that hidden/latent topics to be extracted. LDA find topics from the preprocessed texts.

After LDA extract topics from the pre-processed texts, extracted topics have been saved as feature topics. For our dataset after our model trained successfully and identifies a set of topics for a given dataset, we have selected the top 30 keywords(vocabulary) based on probability and save as CSV file with a topic number, keyword, and probability as column values.

5.2.4 Model Parameter Optimization

LDA model has a number of parameters and some of them has been optimized as follow:

- **corpus**: Set of document vectors or sparse matrix of shape of training datasets. Since topic modeling is unsupervised it need huge collection of corpuses for better training result. We have trained our model over a collection of 5174 texts corpus under six different categories.

- **num_topics**: The number of requested latent/hidden topics to be extracted from the given training corpus. For our dataset as stated in section 5.2 the optimal number of topics is set to 6.

- **id2word**: After texts are tokenized into word level token id2word map from word IDs to exact words. It is used to determine the vocabulary size, as well as for debugging and topic printing.
- **chunksize**: Number of documents to be used in each training chunk. How many texts should be considered in each of training iterations. In our experiment we have set 300 texts per each training iteration.
- **passes**: Number of passes through the corpus during training. How many times does the training pass through the given corpus? The more the value more hidden/latent topics covered. We have set number of passes to 30 for each dataset.
- **update_every**: Number of documents to be iterated through for each update. Set to 0 for batch learning, > 1 for online iterative learning. Used to update the model every number of documents. Our model training is batch learning we have set this value to 0.
- **alpha**: Is Dirichlet distribution over the text. Every topic is given the same alpha value. Alpha can have different values like 0.1, 0.4, 1.0 etc. At low alpha values (<1) most of the topic's distribution samples are near the topic. For really low alpha values means a document may have one topic. For alpha values greater than one the samples come together. Which means large alpha value the topics become uniform. For our experiment we have set alpha value to 0.5.
- **per_word_topics**: Setting this to True allows for extraction of the most likely topics given a word. The training process is set in such a way that every word will be assigned to a topic. Otherwise, words that are not indicative are going to be omitted

After the model successfully trained, it has identified the latent/hidden topics for the health dataset as shown in the figures 5.3 below.

To interpret the result the model is set to identify health topics from the health dataset and print their top 50 most relevant keywords as displayed in figure 5.3.

```
In [8]: runfile('C:/Users/CR7/final_model.py', wdir='C:/Users/CR7')
0:0.026*"ጤና" + 0.016*"በሽታ" + 0.016*"ህክምና" + 0.015*"" + 0.010*"ሆስፒታል" +
0.007*"መድሀኒት" + 0.005*"ደም" + 0.004*"ህብረተሰብ" + 0.004*"ወባ" + 0.004*"ህመም" +
0.003*"ጤና_ጥበቃ" + 0.003*"ክትባት" + 0.003*"አይን" + 0.002*"ምርመራ" + 0.002*"ጤና_ጣቢያ"
+ 0.002*"ካንሰር" + 0.002*"ቀዶ_ህክምና" + 0.002*"ዶ/ር_አማር_አማን" + 0.002*"ሀኪም" +
0.001*"ታላላፊ_በሽታ" + 0.001*"ታካሚ" + 0.001*"መከላከል" + 0.001*"ንቅለ_ተከላ" + 0.001*"ጡት"
+ 0.001*"ቁጥጥር" + 0.001*"ማህበረሰብ" + 0.001*"ቫይረስ" + 0.001*"ጥቁር_አምበሳ_ሆስፒታል" +
0.001*"ረዥም" + 0.001*"ወራት" + 0.001*"ኤድስ" + 0.001*"ወንድ" + 0.001*"ቀናት" +
0.001*"እድገት" + 0.001*"ግል" + 0.001*"ቅድመ_መከላከል" + 0.001*"ስኳር" + 0.001*"ሂደት" +
0.001*"ደም_ግፊት" + 0.001*"አግባብ" + 0.001*"አራት" + 0.001*"ጤና_ኤክስቴንሽ" + 0.001*"ቀላል"
+ 0.001*"ግብአት" + 0.001*"መከላከያ" + 0.001*"ብርሀን" + 0.001*"እንቅልፍ" + 0.001*"ተሳታፊ" +
0.001*"ግብር" + 0.001*"ሀይል"
```

Figure 5.3: Health topics identified by trained LDA model.

The topic is represented as 0.026*" ጤና" + 0.016*" በሽታ" + 0.016*" ህክምና" + 0.010*" ሆስፒታል" + 0.007*" መድሀኒት" + 0.005*" ደም" + 0.004*" ህብረተሰብ" + 0.004*" ወባ" + 0.004*" ህመም" + 0.003*" ጤና_ጥበቃ" + 0.005*" ክትባት" and so on.

Which means the top 50 keywords those participate for this topic are ጤና, በሽታ, ህክምና, ደም, ወባ, ሆስፒታል, ክትባት, ህመም and so on. The weights are indicating the value of the importance of each keyword for the topic. Based on the keywords displayed one can guess that the text is about Health related. More results are presented in annex C.

5.2.5 Neural Word Embedding

We have trained word embedding on the preprocessed texts. We used the Word2Vec approach for representing a word based on its embedding. Word2Vec generates a set of vectors, one vector for each keyword found within the content corpus. We have trained the embedding over 5174 short texts. The output has dimension 300 x V, where V is the size of a unique set vocabulary of a given text. The training was done for each text class independently.

The table 5.1 below shows the co-occurrence of top 5 words of a given keywords trained by the model from each class of texts.

keywords	Co-occurrence word with embedding distance				
በሽታ	መከላከል	መቆጣጠር	ምልክት	ተጠቂ	መራቢያ
	0.96162831	0.95140266	0.9501976	0.94692397	0.94451689
ፊልም	ሰሪ	ዘጋቢ	ተዋናይ	ሜካፕ	ኢንዱስትሪ
	0.93894910	0.92178833	0.9065705	0.89874422	0.88705646
ሰላም	እንዲከበር	ህዝበ	እርቅ	ሽማግሌ	እንዲሰፍን
	0.91373586	0.9050892	0.88413858	0.88402950	0.88249123
ህግ	አንቀጽ	መከራከሪያ	መደንገጉን	ወንጀለኛ	ምህረት
	0.92863100	0.91874849	0.91441309	0.91246187	0.91179275
ኦሊምፒክ	ኦሊምፒያድ	ሜዳሊያ	አይኦሲ	ሜልቦር	ወርልድ
	0.87799715	0.87279725	0.86844289	0.86452972	0.85007387
መተግበሪያ	ሜሴንጀር	መላላኪያ	ዩትዩብ	ማፈላለጊያ	ስካይፕ
	0.96472603	0.962287783	0.96049332	0.95569443	0.9472374

Table 5.1: Word embedding sample

5.3 Evaluation

The clustering result has been evaluated with evaluation metrics of clustering like precision, recall and F-measure. If clustering is done with ground truth labels being present, validation methods and metrics of supervised machine learning algorithms can be used.

Cosine similarity measure is used to calculate the similarity between trained feature and test set texts. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance because of the size (like, if some words appeared more times in one document and less in another) they could still have a smaller angle between them. Smaller the angle, higher the similarity. Which means Cosine similarity is a metric used to measure how similar the texts are irrespective of their size.

The final activity is grouping of related texts using the weighted semantic features of the text document. Text documents are clustered using spherical k-means clustering algorithm in which all text vectors are normalized and cosine similarity measure is applied.

As shown in the figure 5.4 below the trained model is able to predict most of art test set into their correct cluster id. We have 6 cluster id *(1=art, 2=health, 3=other, 4=politics, 5=sport and 6=science and technology)*. For example, total test input text for art dataset set is 101 and out of those 94 has been clustered to art cluster id 1 (true positive); 7 of them clustered in another cluster dataset (false negative); 14 from other dataset cluster to art cluster id (false positive) and 4 dataset has not been clustered.

Depend on the above information we can calculate precision, recall and accuracy for art as follow and the same as for rest dataset:

Precision (P)=true positive/(true positive + false positive)

P=94/(94+14)

=94/108= *0.87*

Recall (R)=true positive/(true positive + false negative)

R=94/(94+7)

=94/101=*0.93*

In figure 5.4 below we have present sample clustering result of art test set with text id and cluster id. More clustering result presented in annex D.

```
In [1]: runfile('C:/Users/CR7/art_testing.py', wdir='C:/Users/
============  Total Input Document  ======== : 101
============  Number of Clusters:   ======== : 6

________________________________________________
Text_ID          |      Cluster_ID (1-6)

________________________________________________
art1               |         N
art2               |         4
art3               |         1
art4               |         N
art5               |         N
art6               |         3
art7               |         2
art8               |         1
art9               |         1
art10               |         1
art11               |         1
art12               |         1
art13               |         1
art14               |         1
art15               |         1
art16               |         1
art17               |         1
art18               |         1
art19               |         1
art20               |         2
art21               |         1
art22               |         1
art23               |         1
art24               |         1
art25               |         1
```

Figure 5.4: Sample Snapshot of art test set clustering result.

The result for each class of short texts clustering result with LDA is shown in the table 5.2 below.

Text class	Number of input document	P	R	Accuracy In %
Art	101	0.87	0.93	89.9
Health	45	0.83	0.87	85
Politics	93	0.94	0.935	93.7
Sport	85	0.94	0.96	94.9
Science and Technology	98	1.0	0.92	95.8
Other	104	0.93	0.8	86
Total	**526**	**0.9**	**0.9**	**90**

Table 5.2: Evaluation Result of LDA model Without Word Embedding.

As shown in the table above the overall performance of LDA model to cluster a total of 526 short texts into six different topics. For each text class we have calculate precision, recall and average accuracy. The total performance of the proposed model is 0.9 of precision, 0.9 of recall and 90% of accuracy. That is from 526 texts 473 of them has been clustered correctly.

In the next table 5.3 we have shown the evaluation of our LDA model trained with word embedding as feature extraction.

From table 5.3 we can notice that word embedding helps the LDA model to predict more accurately than normal LDA. Out of 526 shot texts prepared for testing the LDA model with Word embedding cluster 510 of them correctly which means 97.17% of average accuracy. But our model without word embedding has an average accuracy of 90% as it has depicted in table 5.2.

Text class	Number of input document	P	R	Accuracy In %
Art	101	0.99	0.98	98.47
Health	45	0.9767	0.9333	95.45
Politics	93	0.9887	0.9462	98.42
Sport	85	0.9764	0.9882	98.47
Science and Technology	98	0.989	0.9489	96.8
Other	104	0.99	0.96	97.4
Total	**526**	**0.985**	**0.959**	**97.17**

Table 5.3: Evaluation Result of LDA with Word Embedding

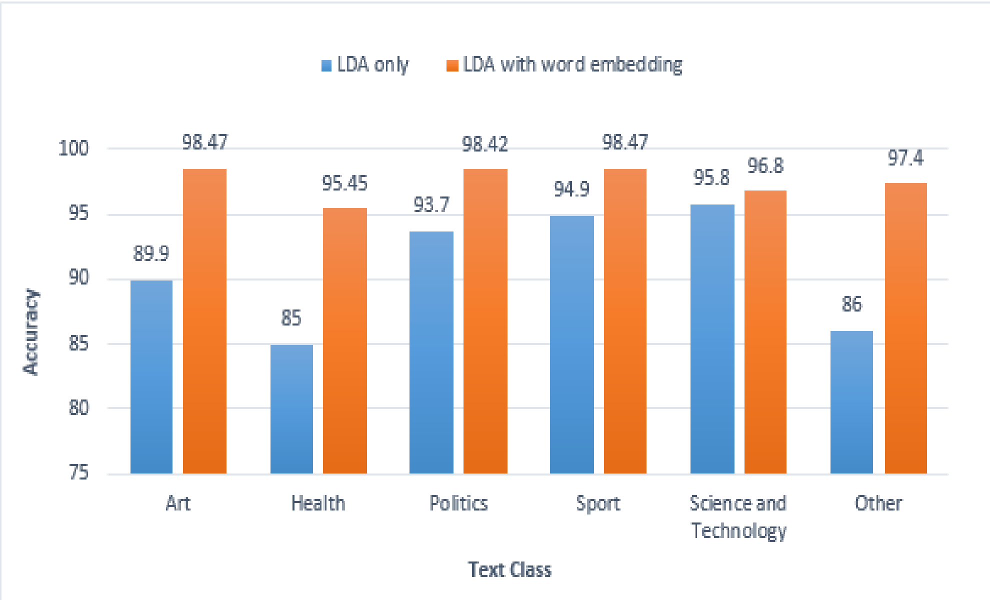

Figure 5.5: Accuracy difference between LDA and LDA with word embedding.

In figure 5.5 we have depicted the accuracy difference between LDA and LDA with word embedding as feature extraction. In the next figure 5.6 we have depicted the accuracy value difference of clustering with and without word embedding.

In general, the LDA model is able to predict short texts into their correct class. But to improve its clustering accuracy a better feature extraction methodology should be employed. As we have seen in the previous section the clustering accuracy of the LDA model increase from 90% to 97.17% when we use word embedding as a feature extraction technique.

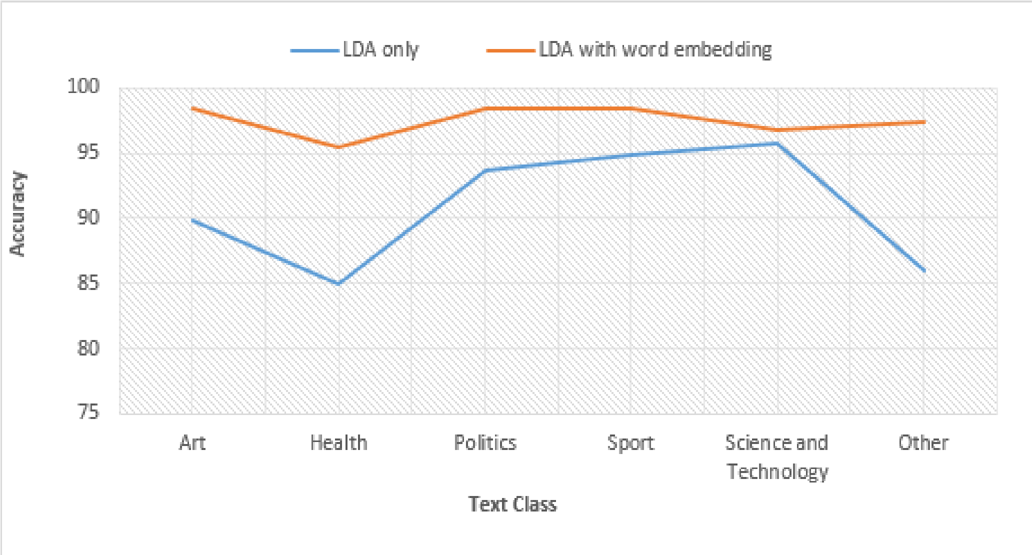

Figure 5.6: Performance curves of clustering with and without word embedding.

Chapter 6

Conclusion, Contribution and Recommendation

6.1 Conclusion

In this thesis work, we have undertaken the clustering of short Amharic texts using topic modeling. To accomplish the study activities like corpus preparation, preprocessing, design, implementation, and evaluation has been done.

The corpus preparation activity performed crawl different local news websites and collect different collection of short news items under different categories. After collecting the news items, the preprocessing module performs stemming, normalization, and stop word removal on the corpus for better machine learning.

The main activity of this thesis work is developing a model for short Amharic text clustering using the topic modeling approach. We used the Latent Dirichlet Allocation (LDA) as a topic modeling tool. LDA assumes a document is a collection of random topics, while the topic is a probability distribution over words. LDA tries to find hidden or latent topics from a given document. In this thesis work, LDA finds out a topic for six different classes of short Amharic texts. To improve better extraction of topics we have combined neural word embedding and topic model to improve the clustering accuracy.

For our experiment, we have identified an optimal number of cluster k is to be six with the Average Silhouette Method.

We have conducted the training experiment on total of 5174 short Amharic texts in *art*, *health*, *politics*, *sport*, *science and technology* and *other* class of topics. We have used 526 texts for testing. Each and every text class recall, precision and accuracy has been recorded and the model has total accuracy of *90%* without word embedding and *97.17%* of accuracy with word embedding as feature extraction as depicted in table 5.2 and 5.3 above respectively. This shows that using word embedding as feature extraction has increase the model accuracy by **7.17%.**

The main tasks undertaken to meet the objective of this thesis work are:

- Identify the requirements for topic modeling and understand the required python libraries.
- In this work we have constructed a generic topic model that can cluster short Amharic text.
- Develop prototype for topic model training.
- Train the proposed model with set of training data.
- Test the developed model with unseen texts.

6.2 Contribution

The contribution of this thesis work is summarized as follow:

- We have developed a model that can cluster short texts into a different group based on the topic.
- This thesis work shows the accuracy difference between clustering texts using topic modeling with and without neural word embedding as feature extraction.
- This thesis work shows that a better feature extraction technique can improve the clustering accuracy.
- This thesis work shows that we can use topic modeling as an approach to perform automatic text clustering.

6.3 Recommendation

Topic modeling is a tool that improves the clustering of texts based on topics. This work tries to address the clustering of short texts using topic modeling. But another researcher effort is needed to make unsupervised text clustering more accurate. There is an assumption that it is possible to cluster too short texts like posts blogs, tweets, comments, opinions into topics beyond sentiment. Future research should consider the following issues:

- Developing an appropriate topic model for short texts because shorts texts do not have enough words they cannot be clustered using a bag of words approach.

- It is better to consider different feature extraction metrics for a better analysis and appropriate data sampling technique.

- For this thesis work we have used LDA as a tool to extract topics and it is unsupervised, but this tool may not be good enough to cluster too short texts. So another research can consider better topic modeling.

- As described in section 5.2 we have set our number of topics to be identified to six, next researches can use more sets of text and a more number of topic.

- Next researchers can consider clustering of even too short texts using feature enrichment techniques.

References

[1] Jain, Abhishek and Lalwani, Surendra and Jain, Suyash and Karandikar, Varun “IoT-based smart doorbell using Raspberry Pi” Springer International Conference on Advanced Computing Networking and Informatics pp. 175—181, 2019.

[2] Y. Wang, E. Agichtein, and M. Benzi, “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.

[3] R. Bekkerman, R. El-Yaniv, Y. Winter, and N. Tishby, “On feature distributional clustering for text categorization,” SIGIR Forum (ACM Spec. Interes. Gr. Inf. Retrieval), pp. 146–153, 2001, doi: 10.1145/383952.383976.

[4] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, “Model-Based Clustering and Visualization of Navigation Patterns on a Web Site,” Data Min. Knowl. Discov., vol. 7, no. 4, pp. 399–424, 2003, doi: 10.1023/A:1024992613384.

[5] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, “Scatter / Gather: Browsing A Cluster-based Large Document Approach Collections to Scatter / Gather,” Sigir’92, pp. 318–329, 1992, doi: 10.1145/133160.133214.

[6] D. R. Cutting, D. R. Karger, and J. O. Pedersen, “Constant interaction-time scatter/gather browsing of very large document collections,” Proc. Annu. Int. ACM SIGIR Conf. Res. Dev. Infofmation Retr., pp. 126–134, 1993, doi: 10.1145/160688.160706.

[7] Anick, Peter G and Vaithyanathan, Shivakumar “Exploiting clustering and phrases for context-based information retrieval” Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval pp. 314—323, 1997.

[8] M. Allahyari et al., “A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques,” 2017.

[9] I. S. Dhillon and D. S. Modha, “Concept decompositions for large sparse text data using clustering,” Mach. Learn., vol. 42, no. 1–2, pp. 143–175, 2001, doi: 10.1023/A:1007612920971.

[10] C. C. Aggarwal, “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.

[11] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” Decis. Support Syst., vol. 74, pp. 12–32, 2015, doi: 10.1016/j.dss.2015.03.008.

[12] Mulualem Wordofa, “Semantic Indexing and Document Clustering for Amharic Information Retrieval”, Unpublished Master’s Thesis, School of Information Science, Addis Ababa University, 2013.

[13] Meron Sahlemariam, Mulugeta Libsie, and Daniel Yacob, “Concept-Based Automatic Amharic Document Categorization”, In Proceeding of the 15th Americas Conference on Information Systems, 2009.

[14] P. Poncelet, M. Teisseire, and F. Masseglia, Data mining patterns: New methods and applications. 2007.

[15] M. Steinbach, G. Karypis, and V. Kumar, “A Comparison of Document Clustering Techniques,” KDD Work. text Min., vol. 400, pp. 1–2, 2000, doi: 10.1109/ICCCYB.2008.4721382.

[16] K. L. Vester and M. C. Martiny, “INFORMATION RETRIEVAL IN DOCUMENT SPACES USING CLUSTERING Department,” Tech. Univ. Denmark, pp. 1–266, 2005.

[17] J. Zhang, Z. Ghahramaniyz, and Y. Yangy, “A probabilistic model for online document clustering with application to novelty detection jian zhangy,” Adv. Neural Inf. Process. Syst., 2005.

[18] Dhillon, Inderjit S and Mallela, Subramanyam and Modha, Dharmendra S “Information-theoretic co-clustering” Proceedings of the ninth ACM

SIGKDD international conference on Knowledge discovery and data mining pp. 89-98,2003.

[19] K. Mrudula and E. K. Reddy, "Hard And Fuzzy Clustering Methods: A Comparative Study Hard and Fuzzy Clustering Methods: A Comparative Study," no. April, 2019.

[20] M. Phil and P. S. G. R. K. College, "Survey on Feature Selection in Document Clustering," vol. 3, no. 3, pp. 1240–1244, 2011.

[21] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," IBM J. Res. Dev., vol. 1, no. 4, pp. 309–317, 2010, doi: 10.1147/rd.14.0309.

[22] P. Singh and M. Sharma, "Text Document Clustering and Similarity Measures," pp. 1–8, 2013.

[23] S. Zhong, "Efficient online spherical k-means clustering," Proc. Int. Jt. Conf. Neural Networks, vol. 5, no. May, pp. 3180–3185, 2005, doi: 10.1109/IJCNN.2005.1556436.

[24] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," Open Comput. Sci., vol. 3, no. 2, pp. 69–90, 2013, doi: 10.2478/s13537-013-0104-2.

[25] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch arid online unsupervised learning," Proc. 7th SIAM Int. Conf. Data Min., pp. 431–436, 2007, doi: 10.1137/1.9781611972771.40.

[26] A. A. Argaw and L. Asker, "An Amharic Stemmer: Reducing Words to their Citation Forms," Proc. 45th Annu. Meet. Assoc. Computational Linguistics, no. June, pp. 104–110, 2007.

[27] M. Gasser, "A Dependency Grammar for Amharic," Work. Lang. Resour. Hum. Lang. Technol. Semit. Lang., 2010.

[28] R. Kramer, “Definite markers, phi-features, and agreement: A morphosyntactic investigation of the Amharic DP,” ProQuest Diss. Theses, p. 346, 2009.

[29] I. Zitouni, Natural language processing of semitic languages. 2014.

[30] F. Murtagh, “A survey of recent advances in hierarchical clustering algorithms,” Comput. J., vol. 26, no. 4, pp. 354–359, 1983, doi: 10.1093/comjnl/26.4.354.

[31] I. Davidson and S. S. Ravi, “Agglomerative hierarchical clustering with constraints: Theoretical and empirical results,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3721 LNAI, pp. 59–70, 2005, doi: 10.1007/11564126_11.

[32] Z. Ying and G. Karypis, “Evaluation of Hierarchical Clustering Algorithms for,” Perform. Comput., pp. 515–524, 2002.

[33] Blei D, Ng Andrew, and Jordan Michael, “Latent Dirichlet Allocation (SHORT),” Nips, vol. 4157698, 2001.

[34] T. Hofmann, “Probabilistic latent semantic indexing,” Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, SIGIR 1999, pp. 50–57, 1999, doi: 10.1145/312624.312649.

[35] F. Esposito, A. Corazza, and F. Cutugno, “Proceedings of the Third Italian Conference on Computational Linguistics CLiC-it 2016,” Proc. Third Ital. Conf. Comput. Linguist. CLiC-it 2016, no. December, 2016, doi: 10.4000/books.aaccademia.1666.

[36] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations ofwords and phrases and their compositionality,” Adv. Neural Inf. Process. Syst., pp. 1–9, 2013.

[37] P. Xie and E. P. Xing, “Integrating document clustering and topic modeling,” Uncertain. Artif. Intell. - Proc. 29th Conf. UAI 2013, pp. 694–703, 2013.

[38] Y. Zhao, G. Karypis, and D. Du, "Criterion functions for document clustering," Proc. Thirteen. ACM Conf. Inf. Knowl. Manag. CIKM 04, pp. 1–30, 2005, doi: 10.1145/1031171.1031225.

[39] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm," Phys. Procedia, vol. 25, pp. 1104–1109, 2012, doi: 10.1016/j.phpro.2012.03.206.

[40] X. Sun, "Textual document clustering using topic models," Proc. - 2014 10th Int. Conf. Semant. Knowl. Grids, SKG 2014, pp. 1–4, 2014, doi: 10.1109/SKG.2014.27.

[41] C. K. Yau, A. Porter, N. Newman, and A. Suominen, "Clustering scientific documents with topic modeling," Scientometrics, vol. 100, no. 3, pp. 767–786, 2014, doi: 10.1007/s11192-014-1321-8.

[42] R. Paredes, J. S. Cardoso, and X. M. Pardo, "Pattern recognition and image analysis: 7th Iberian conference, IbPRIA 2015 Santiago de Compostela, Spain, june 17–19, 2015 proceedings," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9117, pp. 432–440, 2015, doi: 10.1007/978-3-319-19390-8.

[43] C. Li et al., "LDA Meets Word2Vec," vol. 2, pp. 1699–1706, 2018, doi: 10.1145/3184558.3191629.

[44] A. K. Sangaiah, A. E. Fakhry, M. Abdel-Basset, and I. El-henawy, "Arabic text clustering using improved clustering algorithms with dimensionality reduction," Cluster Comput., vol. 22, pp. 4535–4549, 2019, doi: 10.1007/s10586-018-2084-4.

[45] M. Kozlowski and H. Rybinski, "Clustering of semantically enriched short texts," J. Intell. Inf. Syst., vol. 53, no. 1, pp. 69–92, 2019, doi: 10.1007/s10844-018-0541-4.

[46] M. Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya," Conf. Hum. Lang. Technol. Dev., no. April 2011, pp. 94–99, 2011.

[47] T. Demeester, T. Rocktäschel, and S. Riedel, "Lifted rule injection for relation embeddings," EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 1389–1399, 2016, doi: 10.18653/v1/d16-1146.

[48] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," 52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf., vol. 1, pp. 238–247, 2014, doi: 10.3115/v1/p14-1023.

[49] R. Rehurek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," Proc. Lr. 2010 Work. New Challenges NLP Fram., pp. 45–50, 2010.

[50] M. M. Chiang and B. Mirkin, "Determining the number of clusters in the Straight K-means: Experimental comparison of eight options," Proc. 13th Port. Conf. Prog. Artif. Intell., no. April, pp. 395–405, 2007.

[51] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," Neurocomputing, vol. 300, pp. 70–79, 2018, doi: 10.1016/j.neucom.2017.11.077.

[52] `https://fileservice.slidewiki.org/media/images/29/1270.png?filter=Resize-width-368.125`

[53] Internet Usage World Statistics, "Internet and Population Statistics 2017", retrieved from http://www.internetworldstats.com/, Last accessed on October 6, 2020.

Annex: A Unprocessed short text sample used for this work.

የፅንስ ክትትል ለማድረግ የሚረዱ 3 ሺህ 600 ዘመናዊ መሳሪያዎች በሀገር አቀፍ ደረጃ ሊሰራጩ ነው

አዲስ አበባ ፣ ሰኔ 1 ፣ 2011 (ኤፍ. ቢ. ሲ) የነፍሰ ጡር እናቶችንና የፅንስ ክትትል ለማድረግ የሚረዱ 3 ሺህ 600 ዘመናዊ መሳሪያዎችን በሀገር አቀፍ ደረጃ ማሰራጨት ሊጀመር ነው።

የኢትዮጵያ መድኃኒት አቅቦት ኤጀንሲ 3 ሺህ 600 የሚሆኑ የነፍሰ ጡር እናቶች እና የፅንስ ክትትል ለማድረግ የሚረዳ ዘመናዊ መሳሪያ በሀገር አቀፍ ደረጃ ለሆስፒታሎች እና ለጤና ጣቢያዎች ለማከፋፈል ዝግጅቱን አጠናቋል።በቅርቡም መሳሪያዎቹ ለታለመላቸው አላማ ይውሉ ዘንድ ለተመደቡበት ሆስፒታልና ጤና ጣቢያ የሚከፋፈሉ መሆኑን የጤና ሚኒስትሩ ዶክተር አሚር አማን ተናግረዋል። ይህም የእናቶች እና የህፃናትን ሞት ለመቀነስ እየተሰራ ያለውን ስራ ለማጠናከር ከፍተኛ አስተዋፅኦ ያበረክታል ተብሎ ይጠበቃል።

Annex: B Some of Stop Words Used for this work

አመጥ	እንደሌለ	አለመኖር	ለማወቅ	በመሆን	ነገሰ	ማንኛው	የሚያስችል
ማለቱ	ዋና ውስጥ	አስፈጻሚ	ባክዎ	ኢጋድ	አለመሆኑን	ይሄዳል	ወይንም
ተነግሯል	ተናገሩ	እንደነበሩ	ግን ርእይ	ተናግሮአል	በመሄድ	ታውሰዋል	እንዳስታወቀው
ወዲ	ተሰጥቷል	እየተደረገ	የሚደረግ	ይችላሉ	የሚዳርግ	ሲሉም	እንደሚገባው
እንዳለባቸው	ገናማ	የሌለው	በሉ	እንዳለበት	የተነሳ	የመሆን	ብሏል ኖረ
ድረስ	ተያይዞ የት	ተነጋገረው	የሚመጣው	ብለው	ተመልክቷል	መልክ	ቢሆን
ሄደው	የሚደረጉ	ችሏል	ተችሏል	በተለያዩ	የሚሆኑት	የነበራቸው	ይሻላል ነህ
ታይቷል	ምንድነ	ይሄን	ይፋ	ያለውና	ያቀረቡ	አንጻር	ይችላል
እንደተጠበቀ	ያደረገ	ይጠቁማል	አይዘነጋም	ወይም	እንደቻለ	የሚገኝ	አጠናቀዋል
የነበሩ	ትሆን	ቀርቦ እነ	ይበልጥ	በተካሄደው	እንደሆነና	አድርጎ	ካለ ተገለፀ
ዋነኛ ሳልሻ	እናገኛለን	የሚመጣ	የሆነው	ተናገር	ያለ ብሎ	ይጀምራሉ	እንደሚሆን
መኖራቸው	ሆነዋል	ያልሆኑ	ያከናውናል	በሰጡት	አለኝ	ላለ	ምላሽ
በተደረገ	ይልቃል	የተነገረው	ያካትታል	እባኩ	ተናገረብ	ተብለው	የምትሆነው
ከፊል በላይ	ደርሷል	ንዴት	ከነበረው	ወስደው	በማለት	ልብ	አስረድቷል
ወስጥ	የነበረውን	የሚኖር	መናገራቸውን	ያስቸግራል	አስቆጥሯል	ያስፈልገዋል	እንደሚካሄድ
እንጂ	ያስቀምጣል	አይቻልም	አልሁት	ይሁኑ	ያመለክታሉ	በኩል	ይገልፃል
ስለሆነና	እየተባለ	አሏቸው	ባለው	የለውም	እንደሚሆንም	ይመስላል	እንደመሆኑ
ያስታወሱት	ያለበት	በምትሆነው	ሊሆኑ	እኔ	ይናገራሉ	ለሚገኘው	ሆን
ማድረጋቸው	አንዳንድ	አስፈላጊ	አይችልም	ይገመታል	አለች ሆና	ያጠናቀቀው	አክሏል
ተደንግጓል	ይችሉ	እንዲችሉ	ቃለ ያህል	አስገነዘቡ	ወቅት ስነ	ይታወሳል	ይለዋል
ያብራራል	እንደሚገባም	ቀርቧል	ይረዳል	ተጋላጭ	እንደሚገኝ	ለመናገር	እናም
ለውጥ	ተናግሮኗል	እንደማለት	ባደረገው	ወይዘሪት	እንመልከት	ሙሉ	ከሚገኙ
አልቻለም	አንተ	ይኸ	ይልዋል	እንድሆን	መጠቀሙ	ያሳያሉ	ተናገርሁት
ይደርሳል	እንደማይችል	መሆኗን	የሚፈጠረው	በተከናወነው	በተመለከተም	እንዲቋረጥ	እንዳለ
ስለሆነችም	እንዲወገድ	አጠቃላይ	እናንሳ	ከሆነም	ድጋሚ	አሳስበዋል	እየተካሄደ
የሚችለውን	ማድረግ	ባለ	አላት	በዘለለ	ባኩሽ	አለበት	ጭምር
እየተሰጠ	እንደገለፁት	የሚያስችሉ	ይሆኑ	ቢሆኑ	የሚናገሩት	ካለው	እንዲቀጥል
ይጠራል	ነውና	መሆናችን	የሚባል	ካልሆኑ	ምክንያት	ሲነገር	በሆነና
ይገባናል	የሚችል	የሆኑና	ስለሆኑም	እንደሚችል	የሚከሰቱ	እንዳላቸው	ሊያደርግ
ማን	የሚፈጠር	ስለሚሆን	ተጠቁሟል	እንደሚሰራ	እንዳሉ	የሚያደርጉ	ተመር
የሚሆን	ይገልጻል	ይታወቃል	አለው	ስለመሆኑም	ይጠቃሉ	አስቀምጧል	ጠዋል

ወጭ	ለማስፈፀም	አይኖርም	ዋጋ ናችሁ	እንደሆነ	እንዲሰጣቸው	አደረገው	ያቀረበው
ለነበረው	ለሚለው	ይሄ ቢያንስ	አልቻሉም	ይናገራል	ወዘተ	የሚባለው	ያገኛሉ
በተገለፀው	ባለበት	ተገልጿል	ቆይቷል	ይባላሉ	በነበሩበት	አረጋግጧል	መካነ
እንደተገለፀው	ሌላኛ	በነበረበት	ቶሎ	ተናግሯል	ይኖርበታል	ተናገረኸ	ቀደም
ወዳለው	ነገረው	ካሉት	የተደረገው	ወስኗል	ብዙ ለሆኑ	ለመሆኑ	ይቆጠራል
ካለፈው	አቀፍ	ባይቻልም	ሊባል	ሆኑና	ያልሆነ	የተባለው	እርግጥ ነኝ
መኖራቸውን	አቀረበ	ያመለክታል	እሷ ወደ	ስላልሆነ	እንደሚችሉ	ይልቅ	ተናግሮታል
ሆኖታል	ይገኙበታል	ባይሆንም	መካከል	ለማረጋገጥ	ተገቢ	ሆንሁ	መሆን
ገልፀዋል	ሊኖራቸው	እነሱ	ነን	እንዳይሆኑ	እንደሚሉት	ለሚካሄደው	ያምናል
እንዲሰራ	እንደሆን	አስረድተዋል	ይመረጣል	አረጋግጠዋል	ጥቃት	እዚህእ	የሚካሄደው
ነበረ	ወይዘሮ	ሌላ	ይችል	ዘንድ	ረገድ	የሚገባው	ናት
ይገልፃሉ	በመውሰድ	ጠቅሰዋል	ተሰጥቶታል	አመልክተዋል	ካልሆነ	ንዲያው	በተቻለ
እንደማይቻል	ብቻ	መጠን	ይሆናል	ተናገረውስ	በተለይም	እንደሚባለው	ይፈጥራል
እንዲቀንስ	አንስተዋል	ጠቁመዋል	አለችው	እንዳይኖር	እያደረጉ	ይጠፋል	ተደምጧል
ተጨማሪ	መኖሩ	የሚሆነውን	ተናገራት	ውስጥ ሲል	ለመሆን	የነበረው	ለማስወገድ
ተብሏል	እንዳልሆነ	ተገለጸ	ስለሚችል	ያጋልጣል	እንደሚሰሩ	ያስችላል	መሆንን
አስተዋወቀ	ሊያቀርብ	ይደነግጋል	በያዝነው	አድሯል	የሚለውን	ለሚያደር	የሚባሉት
እናገር	እንዲያገኝ	ይጠቁማሉ	የሚችልበት	አለ	እንደሚከተለዉው		እንዳይሆን
ይታወቃሉ	ሊያስከትሉ	እንዳሉት	የተለመደ	ስለሆነም	የተሻለ እና	በሆኑና	የሚሰጡ
አስተዋውቋል	ያሳስባሉ	ያለው	እነሆ	ከመሆኑም	ይመክራሉ	እንዲካሄድ	በመሆኑ

Annex:C Topics identified for Art, Science and Technology and Other text class with our model respectively.

```
In [2]: runfile('C:/Users/CR7/art_compound_token.py', wdir='C:/Users/CR7')
0:0.014*"" + 0.012*"ሙዚቃ" + 0.010*"ፊልም" + 0.008*"ቅርስ" + 0.006*"ደረሰ" + 0.005*"ጽሁፍ" +
0.003*"ስእል" + 0.002*"ግጥም" + 0.002*"ዘፈን" + 0.002*"ዜማ" + 0.001*"ጻፈ" + 0.001*"አርቲስት" +
0.001*"ቅርጽ" + 0.001*"አሳተመ" + 0.001*"ታተመ" + 0.001*"ሰአሊ" + 0.001*"ዳንስ" + 0.001*"ተውኔት" +
0.001*"ምርጥ" + 0.001*"ምስል" + 0.001*"ትርኢት" + 0.001*"ኢትዮጵያ" + 0.001*"ጥበባዊ" + 0.001*"ክሊፕ" +
0.001*"አልበም" + 0.001*"ዘፈነ" + 0.001*"መዝሙር" + 0.001*"አበረታታ" + 0.001*"ዘመኑ" + 0.001*"ሶስተኛ" +
0.001*"ስፖርት" + 0.001*"ሰማ" + 0.001*"ፕሮጀክት" + 0.001*"እንግሊዝኛ" + 0.001*"ሙሽራ" + 0.001*"ተቋቋመ"
+ 0.001*"ተቃራኒ" + 0.001*"ሀፕ" + 0.001*"አሮጌ" + 0.001*"አሳተፈ" + 0.001*"ርእስ" + 0.001*"ተገነዘበ" +
0.001*"ተጋራ" + 0.001*"አደባባይ" + 0.001*"መስፈርት" + 0.001*"አእምሮ" + 0.001*"ስልጣኔ" + 0.001*"ተሞከረ"
+ 0.001*"ፈጸመ" + 0.001*"ሂፕ"
```

```
In [3]: runfile('C:/Users/CR7/tech_compound_token.py', wdir='C:/Users/CR7')
0:0.020*"" + 0.017*"ቴክኖሎጂ" + 0.014*"ኢንተርኔት" + 0.012*"ሞባይል" + 0.008*"ማህበራዊ_ትስስር" +
0.007*"መተግበሪያ" + 0.005*"መረጃ" + 0.004*"ኮምፒውተር" + 0.004*"ፌስቡክ" + 0.004*"አንድሮይድ" +
0.004*"ገመድ_አልባ" + 0.004*"ፈጠራ" + 0.004*"ምርምር" + 0.004*"ኩባንያ" + 0.003*"ስልክ" + 0.003*"ጥናት"
+ 0.003*"ሳተላይት" + 0.002*"አፕሬቲንግ" + 0.002*"ትዊተር" + 0.002*"ሳይንስ" + 0.002*"ኖኪያ" +
0.002*"ጊጋ" + 0.002*"ሜጋ" + 0.002*"ሚዲያ" + 0.002*"ሶፍትዌር" + 0.001*"ጎግል" + 0.001*"ሞባይል_ዳታ"
+ 0.001*"ማይክሮሶፍት" + 0.001*"ሮቦት" + 0.001*"ዲጂታል" + 0.001*"ተንቀሳቃሽ_ምስል" + 0.001*"መሳሪያ" +
0.001*"ድረገጽ" + 0.001*"አይፎን" + 0.001*"ሜሴንጀር" + 0.001*"ሜሞሪ" + 0.001*"ሳምሰንግ" + 0.001*"ፋይ"
+ 0.001*"ፋይል" + 0.001*"ታጣፊ" + 0.001*"ቴሌግራም" + 0.001*"አፕል" + 0.001*"ባትሪ" + 0.001*"ድህረገጽ"
+ 0.001*"መስራች" + 0.001*"ጌም" + 0.001*"ጠፈር" + 0.001*"አፕ" + 0.001*"መለዋወጫ" +
0.001*"ኢንፎርሜሽን"
```

```
In [4]: runfile('C:/Users/CR7/other_compound_token.py', wdir='C:/Users/CR7')
0:0.010*"" + 0.009*"ችግኝ" + 0.007*"ባህል" + 0.005*"ሰላም" + 0.005*"ፓርክ" + 0.003*"ግብርና" +
0.003*"እንስሳት" + 0.003*"ተከላ" + 0.002*"ቱሪዝም" + 0.002*"ቱሪስት" + 0.001*"እርሻ" + 0.001*"አረንጓዴ" +
0.001*"ሰር_ግሮሽ" + 0.001*"ሰብል" + 0.001*"አርብቶ_አደር" + 0.001*"ሰላማዊ" + 0.001*"መስኖ" +
0.001*"መስህብ" + 0.001*"እምቦጭ" + 0.001*"ማሳ" + 0.001*"ሚዲያ" + 0.001*"የሚገኘውን" + 0.001*"መገናኛ"
+ 0.001*"ባለሀብት" + 0.001*"ብርሀን" + 0.001*"ምክክር" + 0.001*"መካከለኛ" + 0.001*"ታላቅ" + 0.001*"ፍች"
+ 0.001*"ፋይዳ" + 0.001*"በቆል" + 0.001*"አስር" + 0.001*"አሸንዳ" + 0.001*"ወንጀል" + 0.001*"ረቂቅ" +
0.001*"ነበር" + 0.001*"ተጠናከረ" + 0.001*"ክ" + 0.001*"ማእከላት" + 0.001*"ወገን" + 0.001*"ጠብቆ" +
0.001*"ሰፊ" + 0.001*"ዜና" + 0.001*"መልሶ" + 0.001*"ለማክበር" + 0.001*"ጉዞ" + 0.001*"ሰብሰቢ" +
0.001*"ዲኤታ" + 0.001*"ጥንቃቄ" + 0.001*"ሀይማኖታዊ"
```

Annex:D Sample snapshot of clustering result for sport and health dataset respectively.

```
In [7]: runfile('C:/Users/CR7/sport_testing.py', wdir='C:/Users/CR7')
============  Total Input Document  ======== : 85
============  Number of Clusters:   ======== : 6

Text_ID          |      Cluster_ID (1-6)

Sport1               |          3
Sport2               |          1
Sport3               |          4
Sport4               |          5
Sport5               |          5
Sport6               |          5
Sport7               |          5
Sport8               |          5
Sport9               |          5
Sport10              |          5
Sport11              |          5
Sport12              |          5
Sport13              |          5
Sport14              |          5
Sport15              |          5
Sport16              |          5
Sport17              |          5
Sport18              |          5
Sport19              |          5
Sport20              |          5
Sport21              |          5
Sport22              |          5
Sport23              |          5
Sport24              |          5
Sport25              |          5
```

```
In [5]: runfile('C:/Users/CR7/testing.py', wdir='C:/Users/CR7')
============  Total Input Document  ======== : 45
============  Number of Clusters:   ======== : 6

_______________________________________________
Text_ID            |      Cluster_ID (1-6)
_______________________________________________
health1                |         N
health2                |         3
health3                |         3
health4                |         3
health5                |         3
health6                |         3
health7                |         2
health8                |         2
health9                |         2
health10               |         2
health11               |         2
health12               |         2
health13               |         2
health14               |         2
health15               |         2
health16               |         2
health17               |         2
health18               |         2
health19               |         2
health20               |         2
health21               |         2
health22               |         2
health23               |         2
health24               |         2
```