ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

UNSUPERVISED TEXT DOCUMENT CLUSTERING USING ENCYCLOPEDIC KNOWLEDGE WITH WORD EMBEDDING

Dessalew Yohannes Bogale

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

(***in Data and Web Engineering***)

Addis Ababa, Ethiopia

October, 2018

Addis Ababa University

College of Natural Sciences

Dessalew Yohannes Bogale

Advisor: **Yaregal Assabie (PhD)**

This is to certify that the thesis prepared by Dessalew Yohannes Bogale, titled: *Unsupervised Text Document Clustering Using Encyclopedic Knowledge with Word Embedding* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science (in Data and Web Engineering) complies with the regulations of the university and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Yaregal Assabie (PhD)		
Examiner: Mulugeta Libsie (PhD)		
Examiner: Mesfin Kifle (PhD)		

Abstract

Digital technologies have made very easy and cheap to generate, store and publish different kinds of data. Thus, almost in every discipline, people are using automated systems that generate information represented in text format in different natural languages. As a result, there is a growing interest towards better solutions for finding, organizing and analyzing these text documents. The effective ways of rearranging the huge amount of text document form later processing, navigating and browsing less complicated, friendly and efficient. Text document clustering is one of the common methods of organizing text documents.

In recent years, Encyclopedic Knowledge (EK) is used in different data mining tasks including text document clustering. Moreover, with the recent advances in machine learning, word embedding is a modern approach for feature learning techniques in natural language documents that is built on the idea that semantics of a word arise simply from its context. Previous works on text clustering do not consider the advantages of using EK with word embedding. In order to improve the performance of text document clustering, this study propose a system that clusters text documents using EK with neural word embedding. EK enables the representation of different related concepts and neural word embedding is used to handle the contexts of these relatedness. During the clustering process, all the text documents pass through pre-processing stages. Then enriched text document features were extracted from each document through mapping with EK and trained word embedding model. Finally, text documents are clustered using the most popular spherical K-means algorithm, that is based on the cosine similarity.

The common evaluation techniques precision, recall and F-measure were used to measure the effectiveness of the proposed system. Amharic text corpus and Amharic Wikipedia data were used for testing. The study shows that the use of EK with word embedding for text document clustering results in 94.95% accuracy showing an average increment of 4.32 % than that of using only encyclopedic knowledge. Furthermore, changing the size of the class has a significant effect on the rate of accuracy and shows that as the cluster size increases the gap in rate of clustering accuracy between using EK with and without word embedding increases. Furthermore, since we do not use any language dependent information in the design process, our system can be applied to other natural language documents having EK.

Keywords: Encyclopedic Knowledge, Neural Word embedding, Concept Based Text Clustering, Feature Enrichment

Dedication

For My Mother (ENATENESH YENIEBAT) （ይቱስ）

Acknowledgments

First of all, I would like to thank God and gratefully acknowledge the help, guidance and support of God in my whole life for giving me the wisdom, strength, support and knowledge in exploring things. Next, I would like to express my gratitude and deeply thankful to my advisor Yaregal Assabie (PhD), who was always there during the process of this thesis work for giving me support, encouragement and continuous advice.

Finally, I would express my special thanks to my former teachers in Jimma University know studying PhD programs; and Addis Ababa university computer science graduate program teachers for their valuable support throughout the course, advise and follow-ups. Then, I would like to thank all of my family members, friends for giving me support and encouragements and also experts working in Amhara mass media agency for giving their time to read and categorize text documents.

Table of Contents

List of Figures iv

List of Tables v

List of Algorithms vi

Acronyms and Abbreviations vii

Chapter 1: Introduction 1

1.1 Overview 1

1.2 Motivation 3

1.3 Statement of the Problem 3

1.4 Objectives 5

1.5 Methods 5

1.6 Scope and Limitations 6

1.7 Application of Results 6

1.8 Organization of the Rest of the Thesis 7

Chapter 2 : Literature Review 8

2.1 Introduction 8

2.2 Amharic Language 8

2.3 Wikipedia 9

2.4 Word Embedding 10

2.4.1 Word2vec 11

2.4.2 GloVe 13

2.5 Text Data Clustering 13

2.5.1 Feature Extraction Approaches 15

2.5.2 Text Feature Weighting 16

2.5.3 Similarity Measures 18

2.5.4 Text Clustering Approaches 20

2.6 Clustering Evaluation Techniques 23

2.7 Summary 25

Chapter 3: Related Work 26

3.1 Introduction 26

3.2 English Text Document Clustering 26

3.3 Chinese Text Document Clustering 29

3.4 Arabic Text Document Clustering 30
3.5 Amharic Text Document Clustering 31
3.6 Amharic Text Document Classification 32
3.7 Summary 34

Chapter 4: Design of Unsupervised Text Document Clustering 36
4.1 Introduction 36
4.2 Design Consideration 36
4.3 Proposed System Architecture 37
4.3.1 Text Preprocessing 40
4.3.2 Neural Word Embedding 42
4.3.3 Encyclopedic Knowledge from Wikipedia 44
4.3.4 Structured Concept Construction 47
4.3.5 Text Feature Extraction 49
4.3.6 Text Feature Enrichment 51
4.3.7 Text Feature Weighting and Clustering 54
4.4 Summary 56

Chapter 5: Experimentation and Evaluation 57
5.1 Introduction 57
5.2 Experimental Procedures 57
5.2.1 Data Collection 57
5.2.2 Tools and Programming Languages 59
5.2.3 Text Data Cleaning 59
5.2.4 Neural Word Embedding 60
5.2.5 Feature Extraction, Enrichment and Weighting 61
5.2.6 Applying Spherical k-means 61
5.3 Evaluation 62
5.4.1 Confusion Matrix 62
5.4.2 Precision, Recall and Accuracy 64
5.4.3 Average Accuracy Values Vs Cluster Size 65
5.4 Discussions 68

Chapter 6 : Conclusion and Future Works 71
6.1 Conclusion 71
6.2 Contribution of the Study 72
6.3 Future Works 73

References 74

Annex A: List of Stop-words 79

Annex B: List of Normalized Alphabets 81

Annex C: List of Amharic Abbreviations from collected corpus 82

Annex D: Sample Output of Clustering 84

List of Figures

Figure 2.1: *Architectures of Word2Vec CBOW and Skip-gram Techniques* *12*

Figure 4.1: *Architecture of Text Document Clustering using EK with Word Embedding* *39*

Figure 4.2*: Tree Like Conceptual Hierarchies from Amharic Wikipedia*..... *46*

Figure 4.3*: Example of Structuring Categorical Concept Relatedness* *48*

Figure 5.1*: Sample Snapshot of Clustering Result* *61*

Figure 5.2: *Accuracy value Difference of Clustering with and without WE*..... *65*

Figure 5.3: *Direction of Average Clustering Accuracy Vs Cluster Size* *67*

List of Tables

Table 4.1: Pairs of Target and Context Words as Training in Windows Size 2 43
Table 4.2: Example of conceptual phrases and words from Amharic Wikipedia.....44
Table 5.1: Manually categorized text documents collected from different sources 58
Table 5.2: Instances of the Distance Values Within Words in Embedding Vector 60
Table 5.3: Confusion matrix for clustering using EK with word embedding results 62
Table 5.4: Confusion matrix for clustering using only EK results 63
Table 5.5: Evaluations of clustering using EK with and without word embedding.....64
Table 5.6: Clustering Result using Different Cluster Size (K). 66
Table 5.7: Accuracy Values of Clustering using Different Cluster Size (K).....66

List of Algorithms

Algorithm 4.1: Training neural network-based word embedding……43

Algorithm 4.2: Structuring categorical concept relationships from Wikipedia data……49

Algorithm 4.3: Steps involved in context feature extraction for a document …….....52

Algorithm 4.4: Generic steps involved in enriched text document feature extraction …...54

Acronyms and Abbreviations

A	Accuracy
BBC	British Broadcasting Corporation
BOW	Bag-of-Words
CBOW	Continuous Bag-of-Words
DLVN	Deep Learning Vocabulary Network
EK	Encyclopedic Knowledge
EM	Expectation Maximization
ENA	Ethiopian News Agency
IDF	Inverted Term Frequency
IR	Information Retrieval
LVQ	Learning Vector Quantization
LWL	Locally Weighted Learning
LSI	Latent Semantic Indexing
NB	Naive Bayes
ODP	Open Directory Project
P	Precision
PD	Pearson Distance
R	Recall
SQL	Structured Query Language
SVD	Singular Value Decomposing
TF-IDF	Term Frequency - Inverse Document Frequency
XML	Extensible Markup Language

Chapter 1: Introduction

1.1 Overview

Recent advancement in storage, networking, data processing, and related technologies has significantly eased the process of generating and collecting large amounts of text data that accommodates huge collections of documents from varied sources, like news portals, analysis papers, books, digital libraries, messages, web sites etc., at extremely high rates and volumes. Due to different contributors across the world, the information is present in varied languages. The increasing amount of documents written in different languages, creates a need to manage that massive amount of varied information. Clustering is one of the main data mining and analysis techniques that deals with organizing a set of objects in a multidimensional space into cohesive groups, for better management and navigation of largescale data [1]. Classification is supervised learning technique used to assign predefined label to instance on the basis of features. In clustering the idea is not to predict the target class as like classification, it is trying to group the similar kind of things by considering the most satisfied condition.

Text clustering is to find out the common representative information from the text documents and grouping these documents into the most relevant groups. Text clustering groups the document in an unsupervised way and there is no label or class information. Clustering approaches have to discover the connections between the document, and then based on these connections the documents are clustered. Grouping of documents into clusters is a basic step in many applications such as indexing, retrieval and mining of data on the web. Given huge volumes of documents, a good document clustering method may organize those huge numbers of documents into meaningful groups, which enable further browsing and navigation to be much easier.

Traditionally, clustering of documents has been regarded as grouping them using predefined classes on the basis of supervised learning techniques. The techniques used mainly use features like words, phrases, and sequences from the documents based on counting and frequency of the features to perform categorization to the predefined classes. However, such results are considered as unsatisfactory since the huge volume of documents may not necessarily reflect the predefined topics. Furthermore, recent trends show the need to shift to unsupervised learning where classes are to be constructed dynamically based on

the semantics of their contents. In such cases, knowledge bases are used to augment unsupervised learning.

Wikipedia is free encyclopedia which has become the largest electronic knowledge repository on the web with millions of articles contributed collaboratively by volunteers [2]. It is much more comprehensive and up to date. In Wikipedia, each article describes a single topic. Equivalent concepts are grouped together by redirected links and each article belongs to at least one category. Wikipedia makes much of its content available for offline analysis through dumps of its database [3]. These database dumps are commonly used as a testbed in the research community and numerous applications, algorithms and tools have been built around or applied to Wikipedia [10]. In the year 2018, there are 299 language editions of Wikipedia including Amharic. A lot of valuable information is being produced in different languages. Increasing amount of text document data may offer lots of useful information to users. Furthermore, the methods of finding information from huge amount of text data needs efficient management. The web search standard defines the result to a user's query as roughly a set of links to the best-matching documents selected out of billions of items available. However, it is challenging to search out the useful information or relevant document from an outsized form of documents. Therefore. drawback of organizing text document is a concern.

Among the methods of document organization is clustering text document. A common approach for text document clustering is by applying statistical techniques for feature selection [4, 5]. This is done by identifying important terms or keywords in the text that best represent the cluster. However, a list of significant keywords, or even phrase will many times fail to provide a meaningful readable label for a set of documents. In many cases, the suggested terms, even when related to each other, tend to represent different aspects of the cluster. Furthermore, encyclopedic knowledge is grounded in human interaction with others and the world around us that is contributed by volunteers. The meaning of text also depends on the aspects of context in which the texts are made.

Therefore, our belief is that the inference abilities of encyclopedic knowledge coupled with the power of a neural word embedding can create more accurate cluster of text documents. Then our main efforts in this work focus on the development of unsupervised text document clustering using the advantage of encyclopedic knowledge with neural word embedding to enhance the clustering results.

1.2 Motivation

Recently, Internet users and text documents written in different natural languages have been dramatically increasing. For example, in year 2016, the number of Internet users in Ethiopia were 4.3% of the population [6]. After one year, in 2017 the number increased to 11.1 %. Furthermore, 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 text documents written in different natural languages are found in the form of books, magazines, newspapers, novels, legal documents, etc. Using a good clustering method, these text documents can be organized into meaningful clusters (groups), which facilitate an efficient browsing and navigation of the text data or efficient information retrieval by focusing on relevant clusters rather than whole text data. The motivation to work on text document clustering arises from the need of accessing and processing the huge collections of text documents more efficiently by organizing effectively.

1.3 Statement of the Problem

Along with the continuously increasing amount of text data availability on the web and different data stores, there is a growing interest in getting better ways of accessing these resources. The effective ways of organizing the huge amount of text document make later processing, navigating and browsing less complicated, friendly and efficient.

Previous works [4, 5] on text document clustering techniques are usually based on the bag-of-words approach. The techniques used in document organization mainly use features like words, phrases, and sequences from the documents based on counting and frequency of the features to perform classification first and clustering to the predefined classes, independent of the context of the term. The bag-of-words approach used on these works [7, 8] is inherently limited, as it can only use pieces of information that are explicitly mentioned in the documents. Specifically, this approach has no access to the wealth of world knowledge possessed by humans. As a result, if two documents use different collections of core words to represent the same topic, they can be assigned to different clusters, even although the core words on the documents are probably synonyms or semantically associated. Therefore, there are words that are not present in a particular document but these words are semantically related with features of a document. Let a document contain words *w1, w2,*

w3, w4, w5, w6 and *w7*, where {*w2, w6} and {w3, w7}* are semantically related words. If the extracted feature of document *d1* consists of say *f1* {*w1, w3, w4, w6}* and text document *d2* consists of *f2* {*w1, w2, w7}* words, then comparing the two documents using only *f1* and *f2* feature do not return good enough result, even though {*w2, w6} and {w3, w7}* are related pairs of words.

A work on classification of text documents [9] uses ontology for classifying only news documents to user predefined category which is not unsupervised learning. Major problem of this approach is that it is usually difficult to design ontology which can cover all the concepts mentioned in a text document collection, especially when the documents to be clustered are from general domain. While replacing original content with ontology terms may cause information loss, especially when the coverage of the ontology is limited. Another problem is that it is difficult to define all categories of different text documents. Furthermore, classification of text document to predefined categories excludes different types of text documents which are unrelated or semantically related to predefined category of documents. The approach proposed in [10] for text clustering uses Wikipedia as background knowledgebase. This approach does not consider contexts of the related concepts within each document that would improve the clustering performance. For instance, the Amharic term "ህመም" would be embedded with {መደንዘዝ, ትውከት, ድካም, ስቃይ} that are descriptive contextual words. The performance of text clustering would be improved if the contexts of extracted feature was considered. Our hypothesis is that the text document clustering abilities of encyclopedic knowledge together with the power of a neural word embedding can create more accurate cluster predictions.

In order to enhance text document clustering by leveraging semantics, two issues need to be addressed: a background encyclopedic knowledge base which can cover the relevant domain of individual document collections as completely as possible; and a suitable text feature extraction method which can enrich the document representation by fully leveraging semantic terms, contexts and relations between the terms. Therefore, this study is an initial attempt to explore the use of encyclopedic knowledge with neural word embedding for clustering text documents. Moreover, unsupervised method of text document clustering by using advantages of encyclopedic knowledge and word embedding for feature extraction that was not included in the previous studies is employed.

1.4 Objectives

General objective

The general objective of this research is to design unsupervised text document clustering using encyclopedic knowledge with word embedding.

Specific Objectives

To accomplish the above mentioned general objective, the following are specific objectives:

- Reviewing and identifying the process of unsupervised text document clustering.
- Collecting and structuring encyclopedic knowledge.
- Collecting text document corpus.
- Designing a model for Amharic text document clustering using encyclopedic knowledge with neural word embedding.
- Extracting descriptive features using Encyclopedic knowledge and neural word embedding.
- Identifying and defining the text document similarity measurement techniques.
- Adopting the appropriate clustering approach.
- Implementing and testing the performance of the system.

1.5 Methods

In order to accomplish the objective, the following system of principles, practices, and procedures will be applied.

Literature Review

Review of literature will be conducted to understand various components of document clustering. Specifically, we will review literature in the area of Amharic language, document clustering techniques, encyclopedic knowledge bases, similarity measurement techniques.

Data Collection

Text document corpus and encyclopedic data will be collected from different offline and online sources. Documents collected will be manually categorized by experts that will be further used for evaluation. These data will be organized and structured in a way that they are easy for experimentation and testing.

Tools

In order to accomplish the objectives of the research, experimentation using available tools and programming will be engaged in the process. PostgreSQL, Python programming language and Java programming language will be used to develop the system.

Testing and Evaluation

To evaluate the performance of the proposed solution, the system will be tested using collected text document data categorized by experts. To evaluate the effectiveness of the proposed system (i.e., clustering results) the most common and basic statistical measures; recall, precision and F-measure will be used.

1.6 Scope and Limitations

This research was conducted to explore the advantage of using encyclopedic knowledge with neural word embedding for unsupervised text document clustering. The scope of the study was to propose and develop an unsupervised text document generic clustering model. In this research work available encyclopedic knowledge from Amharic Wikipedia was used for experimentation. In this study, we considered only textual documents that contain sequence of alphabets without any figure, table, images or any pictorial representations.

1.7 Application of Results

The result of this study can be used as an input for other researches, and possibly applied with the following application area. Text document clustering can be used:

- for text document filtering, pointing to topic-specific processing mechanisms such as information extraction and machine translation.
- to find similar documents matching with the search result document. Clustering is able to discover documents that are conceptually alike compared to search-based approaches.
- for duplicate text content detection: In many applications there is a need to find duplicates in a large number of documents. Clustering is employed for plagiarism detection, grouping of related news stories and to reorder search results rankings.

- for data analytics and recommendation systems: A user can be recommended text documents based on the text document the user has already read. Again this is possible by clustering of the articles, and improving the quality.
- for search optimization: Clustering helps a lot in improving the quality and efficiency of search engines as the user query can be first compared to the clusters instead of comparing it directly to the documents.
- for any organization and application developers which have a large collection of text documents to automatically cluster documents for better management.

In addition, the result of the study will play a role in academics for further researches in the area of using encyclopedic knowledge with neural word embedding.

1.8 Organization of the Rest of the Thesis

The remaining part of the thesis is organized as follows. Chapter Two covers literature review in which different concepts and approaches related to our thesis are presented. Moreover, text clustering and classification, text clustering techniques, Wikipedia (online encyclopedia), neural word embedding, Amharic language, and text document similarity measurement techniques are described. Chapter Three is about works related to our study that are previously done by other researchers in different natural language texts. Chapter Four deals with the design of our system, i.e., unsupervised text document clustering using encyclopedic knowledge with neural word embedding. It presents the general architecture of the system with its basic components; the discussion of the components and their interaction within the system. The algorithms we developed for achieving the goal is presented. Chapter Five focuses on the detail testing and evaluation of the system. It discusses the details about the testing and the results obtained together with their explanations. Conclusions drawn from the thesis result, the contributions of this research work and possible future works are presented in the last chapter.

Chapter 2 : Literature Review

2.1 Introduction

The organization of information into homogeneous groups plays a major role in many fields of research; and clustering is a widely studied grouping technique in the text domain. The method finds numerous applications [13] in document organization, customer division, classification, collaborative filtering, etc. In this chapter, extensive reviews of general concepts on text clustering, clustering techniques, encyclopedic knowledge, text document similarity measurement approaches and text clustering evaluation techniques are presented. The linguistic features of one morphologically complex natural language text, i.e., Amharic is also reviewed. Literature have been reviewed to understand and identify appropriate solution.

2.2 Amharic Language

Amharic is an official working language of Ethiopia and the second most widely spoken Semitic language, next to Arabic [6, 7]. Amharic differs from structure of Semitic languages, especially in syntax. Amharic took the whole Geez alphabet and use it in the writing system. The Amharic alphabet does not have capital and lower case distinctions. It uses a unique script called ፊደል 'fidel' which is conveniently written in a tabular format of seven columns. The first column represents the basic form and the other orders are derived from it by more or less regular modifications indicating the different vowels. Amharic has 34 base characters and total of 435 characters.

Like other Semitic languages, Amharic is one of the most morphologically complex languages [14]. Amharic nouns are the main carriers of information which can be grouped into derived and non-derived nouns. Non-derived nouns are mainly basic or primitive terms that refer to concepts, objects, entities, etc. (e.g. ሰው). On the other hand, derived nouns are formed through morphological processes applied on various word origins. Amharic nouns and adjectives are inflected for number, definiteness, cases (accusative/objective, possessive/genitive) and gender. On the other hand, Amharic verbs are inflected for any combinations of person, gender, number, case, tense/aspect and mood. As verbs are marked for various grammatical units, a single verb can form a complete sentence.

In Amharic language there are different graphemes in which users use in writing interchangeably. For instance, {ሀ, ሕ, ኀ} representing the {h} sound and {ጽ, ፅ} that represent the {ts} sound. Because of this, Amharic text data can contain features that affect processing. Among those, one is orthographic variation (for one meaningful entity, the same word can be written differently in Amharic text data), like ኀይለ ሥላሴ, ኃይለ ስላሴ, ኀይለ ስላሴ, ኀይለ ስላሤ, ኃይለ ስላሤ, ሀይለ ስላሴ, ሀይለ ስላሤ, ሀይለ ሥላሴ, ሃይለ ስላሴ, ሃይለ ሥላሴ, ሃይለ ስላሤ, ኀይለ ሥላሤ, ኃይለ ሥላሤ, ሀይለ ሥላሤ, ሃይለ ሥላሤ, ሐይለ ስላሴ, ሐይለ ስላሤ, ሐይለ ሥላሴ, refers to emperor Haile Selassie (አፄ ኀይለ ሥላሴ); ማህደር, ማሕደር, ማኀደር for Mahedir; ድረ ገጽ, ድህረ ገጽ [4], etc. Amharic texts which refer to the same meaningful element can be written in abbreviation or normal form, example ጠቅላይ ሚኒስትር, ጠ/ሚኒስትር, ጠ .ሚኒስትር for Prime Minster and መስርያ ቤት, መ/ቤት for work office, etc. In addition, for writing numbers in Amharic text data, users might use either the Geez format '፩','፪', '፫'or normal Arabic format '1','2','3', for example, መስከረም ፩ and መስከረም 1. Similarly hyphenated Amharic texts can be written like ወዝ-አደር, ወዝ አደር, ቤተ-እስራኤል, ቤተ እስራኤል. Since there is no standard reference to say that one representation is correct and the other is incorrect, therefore it is used in writing Amharic texts. Amharic language has different punctuation marks used for different purposes. An end of statement is marked with four dots አራት ነጥብ (።) while ነጠላ ሰረዝ (፣ or ፥) is used to separate lists or ideas just like the comma in English and ድርብ ሰረዝ (፤) is used as a semicolon in English. In earlier times two dots (colon) was used to separate words and recently replaced with whitespace. The English question and exclamation marks are also used in Amharic writing system.

Using these language characteristics, users create text documents for different purposes. Amharic text document is a type of computer file that is structured as a sequence of lines of encoded Ethiopic texts and found in online or offline sources presenting a data that forms a report, note, letter, news, etc. In Ethiopia, valuable information being produced are written in Amharic [7]. Recently, there are numerous electronic documents produced and stored in Amharic language. For example, almost all media in Ethiopia including news agencies produce and store huge amount of Amharic text documents.

2.3 Wikipedia

Wikipedia is a free online encyclopedia in which any contributor can create or edit a webpage, and improvements are made within the collaborative environment [3]. Wikipedia has grown to become one of the largest online repositories of encyclopedic knowledge, with millions of articles available for a large number of languages including Amharic [2].

The basic entry in Wikipedia is an article or page, which defines an entity or an event, and consists of a hypertext document with hyperlinks to other pages within or outside Wikipedia. In 2018, there are 299 language editions of Wikipedia. There are 47,110,960 articles in different language editions as of 12 January 2018 [3]. Wikipedia has an underlying model of the knowledge described in its pages and provide the ability to capture or identify information about the data within pages, and the relationships between pages. The role of the hyperlinks is to guide the user to pages that provide additional information about the entities or events mentioned in an article. Categories are used in Wikipedia to link articles under a common topic. The main articles for a category are concepts that are topics, primarily ideas, or abstractions often contrasted with language or reality. Each article in Wikipedia is uniquely referenced by an identifier, which consists of one or more words separated by spaces or underscores. A number of raw database tables in SQL form are available in Wikimedia dump [3]. These are provided usually twice a month which are used as a testbed for researchers [10] and application developers.

Everipedia is wiki-based online English-language encyclopedia which has over six million articles that was launched in 2014 [3]. It is completely open platform where anyone can contribute text, sources, images, and videos by creating a page about something, for a much richer encyclopedia experience. The company was developing a new open source, peer-to-peer wiki network with an incentive structure and a distributed backend hosted within a blockchain. The company claims the new model would make feasible a fully autonomous encyclopedia without the need for advertisements or donations. Everipedia pages are more dynamic in the sense that it is not just only based on what we read. It is also based on the experience, for instance, based on users comment on the articles. Everipedia aims to build the most accessible online encyclopedia, and not be as restrictive as Wikipedia.

2.4 Word Embedding

Word embedding is a modern approach for feature learning techniques in natural language documents. Word embedding is built on the idea that semantics of a word arise simply from its context [15, 16]. It captures both semantic and syntactic information of words, and can be used to measure word similarities, which are widely used in various natural language processing tasks. Neural networks are modern and emerging computational approaches which are revolutionizing current data analytics tasks. Neural networks work with real number preferably with values between 0.0 and 1.0. In order to make use of neural networks

in natural language text processing, we need a way to represent the words as numbers. In word embedding technology, words or phrases from the vocabulary are mapped to vectors of numeric values in which similar words are expected to be close in the vector space [15]. The good feature of word vectors is the contextual similarities between words can be manipulated arithmetically just like any other vector.

For example, all region-related words are very close to each other, for example, "አማራ" (Amhara) and "ኦሮሞ" (Oromo). Even if the words refer to different regions of Ethiopia they are still the topic or concepts of most related text documents. That is, from the point of view of semantic role, they could be considered as related and therefore close to each other in the embedding space.

The most common example to demonstrate the semantic embedding capabilities of word embedding is:

$$vector(\text{" King"}) - vector(\text{" Man"}) + vector(\text{" Woman"}) \approx vector(\text{"Queen "}).$$

There are word embedding techniques like word2vec and GloVe that have shown their advantages in numerous tasks in natural language processing and information retrieval. Word2vec is a technique created by Google that utilizes two different types of model architecture for computing vector representations of words.

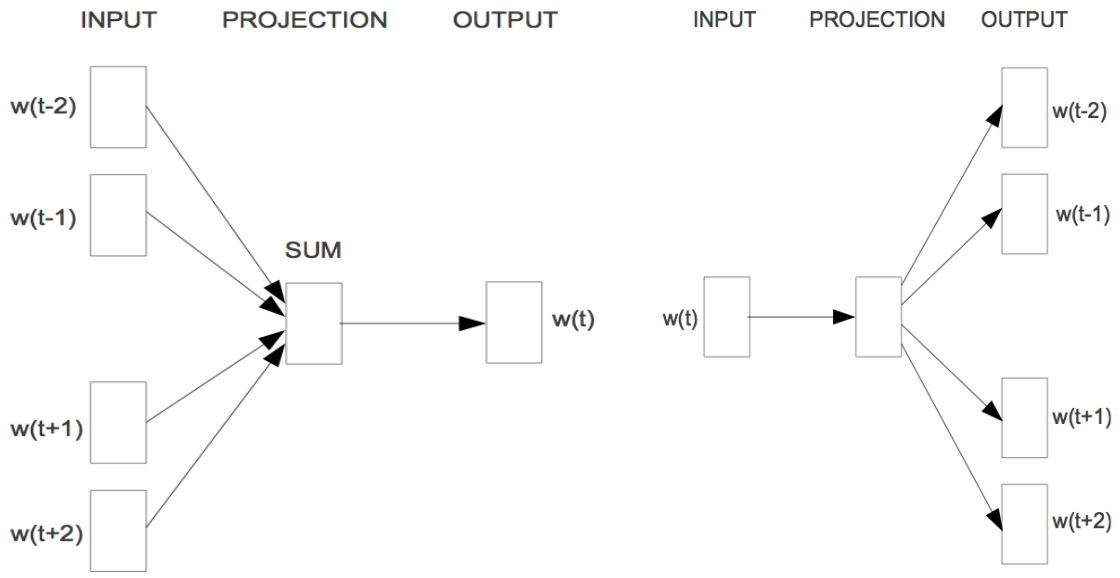
2.4.1 Word2vec

Word2vec is neural word embedding technique that can establish similarities between terms. It is implemented using a two-layer neural network that processes natural language text [15, 16]. Its input is a text corpus and its output is a set of vectors, one vector for each word found in the corpus. The first layer of Word2Vec takes words as one-hot vectors, which is basically a vector of the same length as the vocabulary, filled with zeros except at the index that represents the word we want to represent, which is assigned 1. Then the hidden layer is a standard fully-connected layer whose weights are the word embedding. The hidden layer operates as a lookup table. The output of the hidden layer is just the word vector for the input word. At the end the output layer outputs probabilities for the target words from the vocabulary. Thus word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space. The vectors can be used further into a deep-learning neural network or simply

queried to detect relationships between words. There are two main architectures for Word2Vec, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model.

Continuous bag of words creates a sliding window around current word, to predict it from context (the surrounding words) [15, 16]. After training, these vectors become the word vectors. Using CBOW given a context, we are able to know which word is most likely to appear.

Skip Gram is usually used to predict all surrounding words (context) given a word. With skip-gram, the representation dimension decreases from the vocabulary size to the length of the hidden layer. Furthermore, the vectors are more meaningful in terms of describing the relationship between words. Skip-gram model can capture two semantic vector representations for a single word, for example, it will have two vector representations of አባይ. One for the bank and other for the river.



a) Continuous bag of words b) Skip-gram

Figure 2.1: *Architectures of word2vec CBOW and skip-gram techniques*

As shown in Figure 2.1, the difference between Skip-gram and CBOW is the way the word vectors are generated. For CBOW, all the examples with the target word as target are fed into the networks, and taking the average of the extracted hidden layer [15, 16]. For example, consider the following two Amharic sentences taken from BBC Amharic, ***s1*** "በጤና ተቋማት በየዓመቱ 237 ሚሊዮን ስህተቶችን እየፈጠራሉ" and ***s2*** "በጤና ተቋማት ከሚፈጠሩት ስህተቶች መካከል የተሳሳተ መድሃኒት መስጠት አንዱ ነው". To compute the word representation for the word "ስህተት" using CBOW we need to feed both sentences ***s1*** and ***s2*** into the neural network and

take the average of the value in the hidden layer. Skip-gram only feed in the one and only one target word one-hot vector as input. Using the word2vec architectures we can detect the similarity between word by measuring the similarity.

2.4.2 GloVe

GloVe is word embedding technique used to capture the meaning of one word embedding with the structure of the whole observed corpus using the word frequency and co-occurrence counts as the main measures [16]. Count-based models learn their vectors by doing dimensionality reduction on the co-occurrence count matrix. In this technique, first a large matrix of (words x context) co-occurrence information constructed, i.e., for each word, it counts how frequently we see this word in some context in a large corpus. The number of contexts is obviously large, thus it factorizes the matrix to yield a lower-dimensional (word x features) matrix, where each row now yields a vector representation for each word. Word2Vec and GloVe are implemented in different tools like Gensim library in Python, which can be used to train text document corpus.

2.5 Text Data Clustering

Text analytics is one of the most interesting applications of computing. It involves taking collection of text, converting it into a set of numerical features, and applying a natural language processing or machine learning algorithm on it to derive some insight [17]. Clustering and classification are the two types of machine learning methods which characterize objects into groups by using different features. The task of grouping is highly relevant in today's information age as the massive increase of data to make easy for processing [12]. The processes of clustering and classification appear to be partially similar, but there is a difference between them in context of data mining. The main difference between classification and clustering is that classification is used in supervised learning technique where predefined labels are assigned to instances by properties whereas clustering is used in unsupervised learning that similar instances are grouped, based on their features or properties. When the training is provided to the system, the class label of training tuple is known and then tested, this is named as supervised learning. On the other hand, unsupervised learning does not involve training or learning, and the training sample is not known previously.

Clustering is the process of dividing or grouping the data into a number of groups such that data points in the same group are more similar to other data points in the same group than those in other groups [18]. Rather than defining groups before looking at the data, clustering allows us to find and analyze the groups that have formed naturally. For instance, we could be interested in finding representatives for homogeneous groups, in finding natural clusters and describe their unknown properties, in finding useful and suitable groupings (*searching from useful classes*), in finding unusual data objects (*outlier detection*), grouping of search results, suggestion of related information, recommendation of contents and products, etc.

Gathering the most relevant data for one's need, from the huge collection of data written using varied language is a work of great difficulty [9]. To make it easier, application of **t**ext clustering is used; that is automatic grouping of text documents into clusters, so that documents within a cluster define the similarity between them. Document clustering is an unsupervised machine learning method that separates a large subject heterogeneous collection of text document (corpus) into smaller, more manageable homogeneous collections (clusters). Unsupervised techniques differ from supervised in that they do not require a training sample data or in the case of text documents, the categories are not known in advance. Text document clustering is generally performed simultaneously on a set of documents to arrange them in several groups according to their similarities. The decision whether a text document belongs to a group or another is made dynamically and it is based on the contents of the set of documents. Therefore, clustering does not require the prior definition of grouping, nor the training or predefined rules. Machine learning algorithms prefer well defined fixed-length inputs and outputs but text data are massive for modelling. Thus, typically text clustering involves:

1) Feature extraction from text documents,
2) Feature representation and weighting,
3) Similarity measures between text document features, and
4) Clustering approaches (the ways of grouping based on their similarity values).

These activities are further explained below.

2.5.1 Feature Extraction Approaches

Features are extracted from the text document to represent that particular document for different text mining tasks. Feature extraction from text documents for clustering can be done either **using bag-of words approach** or using **semantic based approaches** [19].

Bag-of-Words Based Clustering

A bag-of-words (BOW) model is a common way of extracting features from text data for use in modeling, such as with machine learning algorithms [4]. It is a representation of text that describes the occurrence of words within a document (considering each word count as a feature). It involves two things, vocabulary of known words and a measure of the presence of known words. It is named as "bag of words" because any information about the order or structure and semantics of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where do words occur and not what does words mean. Bag-of-words approach is used to cluster documents, categorize documents or analyze different corpus.

Bag-of-Words based text clustering is a process of grouping a given document using words selected from the entire text that can describe the content of the document and text is represented as a vector of *v* words weights. Most existing approaches for text clustering represent texts as vectors of words 'bag-of-words' [18]. This text representation results in a very high dimensionality of feature space. In this method, important representative terms (words, phrases) from each text document are selected. A document is clustered based on the similarity of these representative words. In bag-of-word based text clustering, a list of representative keywords extracted from a text document does not describe anything about the semantic relationship between the terms [9]. In most cases, words that are explicitly used in the document are used as keywords for that document. Sometimes the words have a number of meanings or a number of words has the same meaning that would have significant affects in feature extraction.

Semantic based Clustering

Semantics is the study of the meaning of linguistic expressions [3]. Researchers find that the relations between words in natural language texts contribute to understanding the meaning of text. They construct a semantic network in terms of concepts, events, and their relations [8]. Semantic-based text document clustering is a process of grouping a given

document using a set of conceptual semantic representatives that can describe the content of the text document. In this method a set of concepts, 'bag-of-concepts', are extracted from each text document and by using this conceptual representative documents are clustered. The main benefit of the semantic based approach is that it captures and preserves the meanings and associations between words appearing in the document [19]. The documents that are semantically related to each other are grouped into the same cluster and documents that are semantically unrelated are grouped into another cluster. In this approach, semantic knowledge bases such as WordNet, Ontologies [9], Open Directory Project (ODP) and Wikipedia [10] are used to identify the set of concepts appearing within a text document and the relationship between them.

The advantages of semantic based over bag-of-words based clustering are [11, 22]:

- it helps in information and relationship discovery among terms of the documents.
- it helps in retrieving the relevant data efficiently for user queries.
- it can help in semantically relating one cluster to another cluster.
- it helps in generating meaningful clusters and in providing labels to the clusters according to the content of the clusters.

Therefore, it is important to build representations of these text documents which keep their semantics as much as possible and also suitable for efficient similarity calculation.

2.5.2 Text Feature Weighting

After extracting important representative terms, we need to determine how important is a term to a document in a collection of different documents. Evaluating the importance of terms in a document is an important step especially when working with a large textual data analytics because it gives a meaningful insight of what the text document is about. Text Feature weights are calculated by many different schemes which consider the frequency of each term in a document and in the collection as well as the length of the document. A common weighting scheme for terms within a document is to use the frequency of occurrence. The term frequency is somewhat content descriptive for the documents and is generally used as the basis of a representation of weighted document vector. TF-IDF (Term Frequency - Inverse Document Frequency) is a popular and common weighting scheme.

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency - Inverse Document Frequency (TF-IDF) is a weighting scheme that is commonly used in different text analytics tasks and information retrieval to evaluate the importance of a word in a text document [3]. The goal is to model each document into a vector space by retaining information about the occurrences of each word. TF-IDF works by determining the relative frequency of terms in a specific document compared to the inverse proportion of that term over the entire document corpus. This method determines how relevant a given term is in a particular document.

Term Frequency (TF) is the local frequency of a term in the document or the number of times a word/term *t* occurs in document *d* [3]. Terms that are frequent in a particular document are important in identifying what the document is about. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is divided by number of terms in the document as a way of normalization.

- ***TF(t)*** = (Number of times categorical term *t* appears in a document) / (Total number of categorical terms in the document).

However, high frequency of a term alone cannot assure that it is more important than other less frequent words. To correct this TF-IDF method provides a parameter which is the Inverse Document Frequency (IDF).

IDF (inverse document frequency) of a term is the measure of how significant that term across the whole corpus [3]. Here we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

- ***IDF(t)*** = **log** (Total number of documents / Number of documents with term ***t***).

Then, for a term ***t*** in a document ***d***, the weight **W*t,d*** of term ***t*** in document ***d*** is given by computing the following:

- **W*t,d*** = ***TFt,d* * log** (**N**/**DF*t***)

where, TF*t,d* is the number of occurrences of *t* in document *d*, and D*Ft* is the number of documents containing the term *t*.

Therefore, weight of text document doc *i* can be described as [W*i1*, W*i2*, ...W*ij*,....W*in*], where W*ij* is weight value of j the term in the n-dimensional vector space.

2.5.3 Similarity Measures

Similarity measures map the distance or similarity between the representations (descriptions) of two text documents into a single numeric value [21, 22]. The measure reflects the degree of closeness or separation of the target text documents and should correspond to the characteristics that are believed to differentiate the clusters embedded in the data. If this distance is small, it will be the high degree of similarity where large distance will be the low degree of similarity. Similarity measures play an increasingly important role in text related research and applications in tasks such as information retrieval, text classification, document clustering, topic detection, question answering, machine translation, text summarization and others. There are different similarity measures used in development of different applications including text document clustering that results different partitions and also needs different requirements even for the same clustering algorithm. Among the various measures to compute the similarity between text documents, similarity measures which have been frequently used for document clustering are discussed in following sections.

a) Cosine Similarity

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. When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. The cosine similarity of two documents on the vector space is a measure that calculates the cosine of the angle between them [23]. Cosine similarity is one of the most popular similarity measures applied to text documents, such as in numerous information retrieval applications and clustering.

For two documents d_i and d_j the similarity between them is defined as [23]:

$$\mathbf{Cos}(\mathbf{d_i}, \mathbf{d_j}) = \frac{\mathbf{d_i}, \mathbf{d_j}}{||\mathbf{d_i}|| \, ||\mathbf{d_j}||} \qquad (1)$$

When the cosine value is 1 the two text documents are similar, and 0 if there is nothing in common between them.

b) Jaccard Coefficient

For text documents, Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not shared terms [23]. If we have two documents 'a' and 'b', let, terms on a be 't_a', terms on b be 't_b' then the formal definition of Jaccard similarity is [22]:

$$\boldsymbol{SIM\ Jaccard}(\boldsymbol{t_a}, \boldsymbol{t_b}) = \left(\frac{\boldsymbol{t_a}.\boldsymbol{t_b}}{|\boldsymbol{t_a}|^2 + |\boldsymbol{t_b}|^2 - \boldsymbol{t_a}.\boldsymbol{t_b}}\right) \quad (2)$$

The Jaccard coefficient ranges between 0 and 1. It is 1 when $t_a = t_b$ and 0 when t_a and t_b are disjoint, where 1 means the two things are the same and 0 means they are completely different. The corresponding distance measure '$\boldsymbol{D_J}$' is defined as:

$$\mathbf{D_J = 1 - Sim\ Jaccard} \quad (3)$$

c) Euclidean Distance

It is the ordinary distance between two in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text. Measuring distance between text documents, given two documents d_a and d_b represented by their term vectors t_a and t_b respectively with their weight values $w_{t,a}$, $w_{t,b}$, the Euclidean distance of the two documents is defined as [22, 23]:

$$\boldsymbol{D_E}\left(\overrightarrow{\boldsymbol{t}}_{\boldsymbol{a}}, \overrightarrow{\boldsymbol{t}}_{\boldsymbol{b}}\right) = \left(\sum_{\boldsymbol{t=1}}^{\boldsymbol{m}} |\boldsymbol{w_{t,a}} - \boldsymbol{w_{t,b}}|^2\right)^{\frac{1}{2}} \quad (4)$$

where the term set is $T = \{t_1, t_2, t_3 \ldots, tm\}$.

d) Pearson Correlation Distance

This distance is based on the Pearson correlation coefficient that is a measure of the extent to which two vectors are related and calculated from the sample values and their standard deviations [21]. The correlation coefficient (c) takes values from -1 (large, negative correlation) to +1 (large, positive correlation). The pearson distance (*pd*) is computed as *pd* = 1 - c and lies between 0 (when correlation coefficient is +1, i.e. the two samples are most similar) and 2 (when correlation coefficient is -1). Furthermore, more similar two vectors are, the shorter their distance will be. The distance will approach 0 as the correlation goes to 1.

2.5.4 Text Clustering Approaches

Most of the important and commonly used clustering algorithms fit into either hierarchical clustering or non-hierarchical clustering approach [24, 25]. Non-hierarchical text clustering is applied when the goal is to produce text clusters which do not fit in a specific knowledge hierarchy. When a hierarchy is necessary to organize the texts, the hierarchical approach is able to group, for instance, two related clusters inside a major cluster such as taxonomy. Hierarchical text document clustering methods do not create a single clustering result, but the whole hierarchy of clustering.

a) Hierarchical Clustering Approaches

Hierarchical clustering involves creating clusters that build a tree of the data that successively merges similar groups of point data. Hierarchical clustering algorithms are either top-down or bottom-up [25]. Bottom-up algorithms treat each data as a single cluster at the outset and then successively merges pairs of clusters until all clusters have been merged into a single cluster that contains all documents. There are different methods for doing bottom-up (*agglomerative*) clustering [13].

Single linkage method defines the distance between two clusters to be the minimum distance between any single data point in the first cluster and any single data point in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process, we combine the two clusters that have the smallest single linkage distance.

Complete linkage method defines the distance between two clusters to be the maximum distance between any single data point in the first cluster and any single data point in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process, we combine the two clusters that have the smallest complete linkage distance.

Average linkage method defines the distance between two clusters to be the average distance between data points in the first cluster and data points in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest average linkage distance.

Centroid method defines the distance between two clusters as the distance between the two mean vectors of the clusters. At each stage of the process we combine the two clusters that have the smallest centroid distance. Top-down clustering (Divisive) requires a method

for splitting a cluster [24]. In this method first we assign all of the observations to a single cluster and then partition the cluster to two least similar clusters. Then, we proceed recursively on each cluster until individual data are reached.

b) Non-hierarchical Clustering Approaches

A non-hierarchical approach generates some categories by partitioning a dataset [24, 25]; giving a set of non-overlapping groups having no hierarchical relationships between clusters. In a non-hierarchical method, the data are partitioned into a set of K clusters and this may be a random partition or it may be a partition based on a first guess at seed points which form the initial centers of the clusters. Then data points are iteratively moved into different clusters until there is no reassignment possible. There are a number of techniques for non-hierarchical clustering, but we have described K-means which is widely used in text document clustering.

i. K-means Clustering

K-means is a famous unsupervised clustering algorithm used to organize the data, that is used when we have unlabeled data (data without defined categories or groups) [24, 26]. The basic algorithm finds groups in the data, with the number of groups represented by the variable K. K-means algorithm works iteratively to assign each data point to one of K groups based on the features that are provided [25]. There are many methods of estimating K but there is no method for determining the exact value of K. One simple rule for deciding the optimum number of clusters (K) to have is $K = \sqrt{N/2}$.

Steps for basic k-means clustering algorithm is given below:

Let $X = \{x_1, x_2, x_3, \ldots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \ldots, v_c\}$ be the set of centers.

1) Randomly select *c* cluster centers.
2) Calculate the distance between each data point and cluster centers.
3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
4) Recalculate the new cluster center using:

$$Vi = \left(\frac{1}{Ci}\right)\sum_{j=1}^{Ci} Xi \qquad (5)$$

where, c_i represents the number of data points in the i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.
6) If no data point was reassigned then stop, otherwise repeat from step 3.

The results of the K-means clustering algorithm are [26]:

- The centroids of the *K* clusters, which can be used to label new data.
- Labels for the training data (each data point is assigned to a single cluster).

Spherical k-means is the most popular method of clustering text data in which the algorithm takes cosine similarity between data [27]. In grouping (clustering) process, each cluster mean vector is updated, only after all document vectors being assigned, as the (normalized) average of all the document vectors assigned to that cluster. Spherical k-means algorithm is given as:

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

K-means algorithm does not depend on the order.

ii. Density Based Clustering

Density based clustering refers to unsupervised learning method that identifies distinctive groups/clusters in the data [28], based on the idea that a cluster in a data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density. The data points in the separating regions of low point density are typically considered noise/outliers. Clusters are regarded as regions in the data space in which the objects are dense, and which are separated by regions of low object density (noise). Density-Based Spatial Clustering (DBSCAN) is the basic density-based clustering algorithm that can be used to find flat clusters. The density based clustering result is influenced by parameter, because in most cases it needs to define the neighborhood density threshold and radius.

Steps for basic density based clustering algorithm is given below:

1) Choose a random point ‘r’ radius.
2) Calculate all data points which satisfy density from ‘r’ with respect to radius and density, i.e., minimum points.
3) If ‘r’ is a core point, then it forms a cluster.
4) If ‘r’ is a border point, no data points reach the density from ‘r’, then the algorithm goes to the next data point in the space.
5) Repeat the process until all the points in the space are covered.

Density based clustering algorithm [29] can create nonlinear set of clusters and it is not sensitive to noise. Density based clustering is the second best clustering method after k-means and the complexity is low.

iii. Expectation Maximization Based Clustering

Expectation maximization (EM) is a well-known iterative clustering method for learning probabilistic categorization model from unsupervised data [3]. The expectation maximization clustering method initially assumes random assignment of examples to categories. It uses the following two steps until convergence: Expectation (E-step) where each object is assigned to the centroid such that it is assigned to the most likely cluster: Compute probability for each example given the current model, and re-label the examples based on these posterior probability estimates. Maximization (M-step): Re-estimate the model parameters from the probabilistically re-labeled data. where the model (centroids) are recomputed.

2.6 Clustering Evaluation Techniques

The final goal of clustering is attaining high intra-cluster similarity (similarity of text documents within a cluster) and low inter-cluster similarity (similarity of text documents from different clusters). When comparing a cluster solution, we can consider 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 [30].

Basically when we consider **precision** and **recall** from information retrieval concept, each cluster is considered as if it were the result of unsupervised clustering and each class as if

it were the desired set of documents for the category. Formally defined as follows [34]; which are widely used to evaluate the performance of unsupervised learning algorithms.

$$\boldsymbol{Recall}\ (\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{n_{ij}}}{\boldsymbol{n_j}}; \quad \boldsymbol{Precision}(\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{n_{ij}}}{\boldsymbol{n_i}} \tag{6}$$

where $\boldsymbol{n_{ij}}$ is the number of documents with class label *i* in cluster *j*, $\boldsymbol{n_i}$ is the number of documents with class label *i* and $\boldsymbol{n_j}$ is the number of documents in cluster *j*.

Thus, the F-*measure* [33, 34] for cluster *i* and class *j* is defined as:

$$\boldsymbol{F}(\boldsymbol{i},\boldsymbol{j}) = \frac{\mathbf{2} * \boldsymbol{Recall}\ (\boldsymbol{i},\boldsymbol{j}) * \ \boldsymbol{Precision}(\boldsymbol{i},\boldsymbol{j})}{\boldsymbol{Recall}\ (\boldsymbol{i},\boldsymbol{j}) + \ \boldsymbol{Precision}(\boldsymbol{i},\boldsymbol{j})} \tag{7}$$

Accuracy is defined as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0.0. The higher f-measure is the higher accuracy of cluster.

Other measurement method related to the internal quality of clustering is ***entropy*** measurement and it is defined as [34]:

$$\boldsymbol{E_j} = -\sum_{\boldsymbol{i}} \boldsymbol{P}(\boldsymbol{i},\boldsymbol{j}).\boldsymbol{LogP}(\boldsymbol{i},\boldsymbol{j}) \tag{8}$$

where, P (i, j) is probability that a document has class label i and is assigned to cluster j.

Thus, the total entropy of clusters is obtained by summing the entropies of individual clusters weighted by the size of each cluster. The lower value of entropy, the higher quality of cluster.

Purity is an external evaluation technique of cluster quality. The *purity* measure evaluates the coherence of a cluster, that is, the degree to which a cluster contains documents from a single category [30]. Purity can be interpreted as the classification rate under the assumption that all samples of the cluster are predicted to be members of the actual dominant class for the cluster. High purity can be easily achieved when the number of clusters is large; purity is 1 if each document gets its own cluster. External measures are related to how representative are the current clusters to true classes. The purity and entropy measure the ability of a clustering method to recover known classes (for example, if one knows the true class labels of each corpus).

2.7 Summary

This Chapter explained text document clustering which is the process of grouping similar data into different groups. Moreover, it is the partitioning of a data set into subsets, so that the data in each subset is according to some defined similarity measure. Mostly, clustering deals with unsupervised data; thus, unlabeled whereas classification works with supervised data; thus, which are labeled. This is one of the major reasons why clustering does not need training sets while classification does. Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. The clustering methods differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. Non-hierarchical clustering, on the other hand, attempts to directly decompose the data set into a set of disjoint clusters. Clustering algorithms are useful for exploring data. K-means is especially useful and commonly used. The most common clustering approaches are discussed in this chapter including concept-based and bag-of-words-based clustering techniques. Similarity measures play an increasingly important role in text related research and applications. The similarity measure is the measure of how much similar two data items are. Similarity measure in a data mining context is a distance with dimensions representing features of the data item. If this distance is small, it will be the high degree of similarity where large distance will be the low degree of similarity. This chapter also reviewed literatures on commonly used text similarity measurement techniques. The Cosine similarity, Euclidean and Jaccard similarity measures were presented. Wikipedia has grown to become one of the largest online repositories of encyclopedic knowledge, with millions of articles available for a large number of languages including Amharic. Most of the valuable information produced in Ethiopia from increasingly amount of media are written in Amharic. To measure the quality of clustering results, evaluation measure of clustering is needed. We have also reviewed and discussed the most common text clustering evaluation techniques. Evaluating the clustering result shows how well the clustering is performed and how good are the produced clusters.

Chapter 3: Related Work

3.1 Introduction

Text document clustering has been heavily researched and studied for improving the precision in information retrieval systems [1] and for text analysis to automatically generate groups or clusters of text documents. Different approaches for different natural language text documents to solve the problem of text document organization (clustering and classification) have been proposed by many researchers, for instance, for Amharic, English, Chinese, Russian, Ukrainian [33], etc. This Chapter presents the previous research on natural language (Arabic, Amharic, English and Chinese) texts that are related to our study. Among many text document clustering works done, we have chosen the most relevant ones which are related to our work and done on different natural language text documents.

3.2 English Text Document Clustering

As we have discussed in the previous chapter, clustering can be done either using bag-of words approach or using semantic based approaches of feature extraction with text data [19]. More recent researches done using these techniques on English text documents are discussed on the next sections.

a) Bag-of-Words based Approach

Reddy *et al.* [34] developed clustering algorithm based on frequent word sequences that can provide valuable information about the text documents and implemented in Java. Users may not get what they want from the top retrieved documents on the list by search engines. Thus, the main aim of the paper was to increase the precision of the retrieval result by clustering before presenting to the user. In this work, apriori algorithm for frequent item set mining and association rule learning was used. Based on the work using features like sequential relationship, frequent word sequencing and word meanings; clustering algorithm attains good performance and high speed results at the same time. As stated in this work, if the knowledgebase was used during feature representation, the clustering result would be more enhanced. More similar to this, Kumar *et al.* [35] also attempted frequent term based text document clustering. Based on their work each pair of term frequency vector was compared to find out the similarity value between every two corresponding documents; and similarity matrices minimum-match, maximum-match and average-match are generated.

They used the similarity matrices to cluster and explored the relations between number of frequent terms, similarity measurement and evaluation methods. According to the work, internal measure objective function is used with the goal of maximizing intra-cluster similarity (similarity between documents within a cluster) and minimizing the inter-cluster similarity (similarity between documents from different clusters). But for external quality measures external knowledge about the data was required. The research work was based on term frequency and does not consider the semantics and relationships between the extracted terms.

Yi *et al.* [8] attempted text clustering using deep-learning vocabulary network. They presented a graph-based approach for text clustering, named deep learning vocabulary network (DLVN). The vocabulary network was constructed based on related-word set, which contains the co-occurrence relations of terms. In this work, the edges of vocabulary network were used to represent the relations between words or terms and extract features of text documents in terms of related-word set. Frequent item-set algorithm was used to obtain co-occurrence relations between words or terms; association rules learning was used to obtain relations between words; and employed deep learning for dimensionality reduction. In this study an edge was added to the vocabulary network by considering the semantic and relatedness information among terms. Page-rank scores were used to obtain the importance of feature vectors instead of the term frequency and Deep-Learning Single-Pass algorithm for clustering. PageRank algorithm was used to count term frequency not only by classic metrics of TF and TF-IDF but also by term-to-term associations. The idea of this technique was that documents which share a set of words that appear frequently are related, and this was used to cluster documents. But this does not infer that these words are exactly related to each other; all related words do not appear together in text documents and also the context of relation between words is not considered.

b) Semantic based Approach

Huang *et al.* [18] proposed a system for clustering documents with active learning using Wikipedia as a background knowledgebase. This study was to explore the semantic knowledge in Wikipedia for grouping of documents and enabling the automatic clustering of similar documents. They used supervised approach using active learning. In this work, first Wikipedia concepts were utilized to create a concept-based representation of a text document. After identifying candidate phrases in the given document, they mapped them

to Wikipedia articles. Furthermore, selection was based on analyzing the major concept groups representing the major threads in the given document collection. As mentioned in [18] the two concepts are considered to be neighbors if the semantic relatedness between them was not less than a pre-specified threshold and eliminate concepts whose value falls below a certain threshold. First, clustering of the most frequent concepts was done according to their semantic relatedness with all the concepts. For each concept cluster, related documents were retrieved and ranked based on their weight for the concept cluster. The technique used considers concept clustering first then after ranks documents based on the cluster. This approach does not consider contextual relations of the concepts within each document to be clustered and it clusters concepts then ranks the text documents.

Hu *et al.* [10] used Wikipedia concepts and categories for text document representation. They proposed two approaches for mapping concepts to the documents. The first approach was called exact-match that is a dictionary-based approach. It maps the topical terms present in the documents directly to Wikipedia concepts. The second mapping approach was relatedness match in which instead of mapping Wikipedia concepts to each document directly, this approach builds the connection between Wikipedia concepts and each document based on the contents of Wikipedia articles. Based on the work English texts documents were clustered based on a similarity metrics. The proposed framework was evaluated on three English text datasets and used both agglomerative and partition clustering for experiments. Based on the results, it was explored that in agglomerative clustering method, enriching document representation with Wikipedia concepts and categories by both exact-match and relatedness match can significantly improve the clustering performance. This approach did not consider contextual relations of the extracted concepts within each document to be clustered. Performance would be more improved if the contextual relationships between extracted feature was considered.

Yang *et al.* [36] proposed an approach for mining hidden concepts based on short text clustering using Wikipedia as background knowledgebase. This work was based on increasingly available short texts in social networking platforms. In this work, Wikipedia concepts were identified in documents and these documents were enriched by searching related concepts. After texts were enriched with Wikipedia knowledge, clustering using bisecting k-means algorithm based on topics was performed. This work explored that using Wikipedia as a resource for enriching texts can improve performance in community mining.

This approach did not consider contextual relations of the concept within each document that would improve the clustering performance. If the contextual relationships between extracted features were considered, then the performance of the system would be improved.

Li *et al.* [37] proposed a novel framework named document concept vector for cross-domain text classification. In this work the raw document was first transformed into a conceptualized document which consists of a set of concepts. After that, the conceptualized document was transformed into a document vector through the neural network. This vector was used as the concept level feature of the original document. In this study, entities were recognized from a document through backward maximum matching algorithm and concept of the entity was determined by mapping a taxonomy knowledge base. A neural network is used for training these two kinds of vectors. After training, the document vectors could be regarded as the concept level features of the original document that were used to predict the class labels of the documents. This work categorizes text documents, i.e., supervised learning by training text documents. Training needs enough documents for each predefined category that has the limitation of defining preparing training set for each class. This approach did not consider contextual relations of the concept sets within each document to be clustered.

3.3 Chinese Text Document Clustering

Yao *et al.* [32] attempted Chinese text clustering based on k-means algorithm. Vector space model was used that maps each document to a point of vector space. Then document category was decided by words or vocabularies and their frequency. This study presented an improved method of k means algorithm for Chinese texts, in which the idea was selecting documents more similar to the cluster center to calculate average value as new center when updating cluster center. Average similarity of one cluster was used as a parameter, and multiplied it with a modulus value defined to get the similarity threshold value, the text documents whose similarity with the original cluster center was greater than or equal to the threshold value were collected as a candidate collection, then updated the cluster center with center of candidate collection. When compared with the original K-means algorithm (not improved K-means) the time complexity was not affected while clustering results were improved. New center of a cluster was easily effected by isolated text and proved that the algorithm is correct and effective by experiments. As discussed in

the study, the overall result was not satisfactory because the approach used did not consider the meaning and relationships of the words, named as bag-of-words approach.

Han *et al.* [47] carried out a research on Chinese document clustering based on a datamining tool WEKA. In this study the preprocessing functions of WEKA were used to clean documents. Then text files were converted into ARFF form. Strings were converted to sparse word vectors. In the process of String to vector conversion the term weight and feature selection functions were used. Basic K-means Clustering algorithm was applied. When wordsToKeep ranged from 20 to 250, F-measure was between 76.26% and 82.03%, recall was between 76.58% and 81.33%; precision was between 79.74% and 84.11%. When wordsToKeep was larger than 400, F-measure and precision began to decrease significantly. However, recall increased gradually. WEKA was sensitive of sparse data, if the feature-document matrix is fairly sparse, a large number of documents can be divided into one cluster. As a result of high recall but low F-measure and precision. Finally, when wordsToKeep was 5000, 18819 distinct features were selected in total, account for 77.0% of all features. While wordsToKeep was larger than 5000, the clustering accuracy was nearly the same. This work was not semantic based clustering or it was not based on the semantics of text documents.

3.4 Arabic Text Document Clustering

Froud and Lachkar [41] attempted Agglomerative Hierarchical Clustering Techniques for Arabic Documents. The Arabic language has a complex morphology and is highly inflected like that of Amharic language. In this study bag-of-words approach was used to represent documents. Hierarchical clustering using seven linkage techniques with different distance functions and similarity measures, such as the Euclidean Distance, Cosine Similarity, Jaccard Coefficient, and the Pearson Correlation Coefficient was tried. For the testing, the experimentation was done three times: without stemming, with stemming using the Morphological Analyzer and using different similarity measures. The goal was to decide which are the best and appropriate techniques to use for producing consistent clusters for Arabic Documents. As presented in this work, the conclusion was, (1) for the agglomerative hierarchical algorithm, the use of Ward function as linkage techniques yield good results; (2) Cosine Similarity, Jaccard and Pearson Correlation measures perform better relatively to the other measures; and (3) The tested documents clustering technique perform well

without using stemming. The bag-of-words approach did not consider the semantic relationships between extracted features.

Al-Anzi and AbuZeina [39] carried out a research on Categorization for Arabic Text Using Latent Semantic Indexing (LSI). In this study, Latent Semantic Indexing (LSI), singular value decomposing (SVD) method for feature representation and clustering techniques to group similar unlabeled document into pre-specified number of topics were used. The generated groups are then categorized using a suitable label. For clustering, EM and K-Means algorithms were used. For experimentation, they created a corpus that contains 1000 documents belonging to 10 different categories. From the corpus, a term-by-document matrix was created using only term counts. In this work, class topics were predefined (supervised learning) and contextual relations between features was not considered.

3.5 Amharic Text Document Clustering

Mulualem Wordofa [5] carried out a research on semantic indexing and document clustering for Amharic information retrieval. This work was based on the term frequencies or keyword based approach. In this work, a document summary for each cluster containing the distinct terms whose frequencies are high is prepared after preprocessing of the documents. The author used K-means partitioning algorithm and mode based cluster representative selection has been used.

Abegaz Yalemsaw [40] on document clustering for Amharic texts explored the advantage of document clustering to improve information organization and retrieval performance of documents in Amharic language. The author collected different news text document corpus, preprocessed and stored them in a vector with their corresponding term frequency and inverse document frequency. Furthermore, frequent item set hierarchical clustering algorithm was used to organize documents. In this study hierarchical document clustering demonstrated improvement in the performance of Amharic information retrieval systems.

Mulualem Wordofa [5] and Abegaz Yalemsaw [40] used keyword based approach that does not consider the semantic relationships among words. Most of the morphological variations in Amharic occur in the verb that will probably result in high term frequency value, while the nouns or entities are the main carriers of information in Amharic language relevant for a clustering task. Thus term frequency vectors keep the dimensionality of the data very high. Semantically poor representation of text documents without considering the meaning

results in poor quality of clusters. If two documents use different collections of core words to represent the same topic, they may be falsely assigned to different clusters due to lack of shared core words. Recently a common way to solve this problem is to enrich document representation with background knowledge.

3.6 Amharic Text Document Classification

Most of the research works on Amharic text document organization are done on classification by defining specific sets of classes. These works are done based on bag-of-words and semantic based approach.

a. Bag-of-Words Based Approach

Due to the morphological complexity and structural difference of Amharic language, unsupervised clustering of Amharic text documents has become difficult to carry out, and most of the research works are done on classification by defining specific sets of classes. Samuel Eyassu *et al.* [41] proposed an attempt to mine Amharic text from the web and then discussed several classification experiments that were performed on the compiled corpus. In this work, document weighted matrix term vector was used for training. Three groups of experiments were done. In the first two they used the self-organizing map model of artificial neural networks for the task of classifying a collection of Amharic news items. According to this work, weighted matrix was generated from the original document term matrix using the log-entropy weighting formula. The documents were classified based on the training data patterns.

The other study in Amharic news document domain is the one done by Yohannes [41]. The objective of the study was to develop or adopt processing tools for Amharic text classification and evaluate the performance of selected classifiers for Amharic text classification tasks. Yohannes Afework focus was on developing a document pre-processing scheme which facilitates efficient automatic classification of Amharic documents. The works above used keyword-based approach that uses a long list of words as vector space to categorize a given document to a predefined class. This approach is often unsatisfactory for a couple of reasons: first, it keeps the dimensionality of the data very high, and second, it ignores Semantics or important relationships between terms.

Worku Kelemework [43] proposed a neural network approach for Amharic text news classification. In this study, Ethiopian news agency data was used for training using

learning vector quantization (LVQ) method. Term frequency weighting (TF) and TF with IDF (inverse document frequency) weighting methods are used and compared the results of the two methods. Based on experimentation done, the study explored that TF weighting scheme is better in accuracy than TF-IDF weighting scheme. This work was not based on the semantics of Amharic text documents instead he used keyword or term weighting representation are used.

Seffi Gebeyehu and Vuda Sreenivasa Rao [42] proposed an algorithm for learning from labeled and unlabeled documents. This work was based on the combination of Expectation-Maximization (EM) and two classifiers: Naive Bayes (NB) and Locally Weighted Learning (LWL). Term weighted vector feature representation was used without considering the semantic relationships among terms. In this work, first they used EM clustering algorithm to group classes to clusters of the mixture document so that both labeled and unlabeled documents will be clustered to the predefined classes. Then text classifying algorithms are used to predict the documents to their predefined categories. Based on the study, class topics were predefined (supervised learning) and contextual relations between features did not considered.

Abraham Hailu and Yaregal Assabie [44] proposed a system that classifies Amharic documents based on the frequency of item-set obtained after preprocessing of Amharic text documents. In this study, frequent item-sets were used to generate terms linked with categories and thus item-sets are used as input for training phase. Based on training, the category of a new document was supposed to be predicted. Extended version of apriori algorithm was used. Alemu Kumilachew *et al.* [45] explored the use of hierarchical structure for classifying Amharic new text documents. In This work, support vector machine which is a method for supervised learning was used. Furthermore, the effects of the number of categories, number of top features on flat classification and hierarchical classification was discussed. Depending on their experimental results they conclude that hierarchical Amharic text classification approach shows good result in classifying documents into their predefined categories. This work is [44] based on term-frequency which do not consider the semantic relationships among words. A word or term may frequently occur in one document and its synonym or4other representative word may occur in different documents, thus it cannot accurately represent the meaning of documents. It was based on learning from training data that needs the correct input training text data so

that learning algorithm finds patterns in the training data that map the input text document attributes to the target. Defining every categorical text document training sets is difficult and it would not handle a new document type that do not included during training so that unsupervised clustering is needed.

Semantic based Approach

Meron Sahlemariam *et al.* [9] attempted to look into the techniques of automatic classification of Amharic text documents to predefined categories which is not unsupervised learning. In this work, for categorizing a given document into a predefined class, the document passes through the pre-processing and classification steps. In order to classify a given document, the knowledge that contains concepts in the news domain was represented using prepared ontology. After the representation of domain concepts, document representative terms are extracted from the document as index terms. Using thus index terms that are extracted from the document and ontology knowledge base, a given document is classified into predefined categories. The limitations of concept based approach by designing ontology [9] is: (i) it is classification to predefined categories of Amharic news text documents only. There are different kinds of Amharic text documents. For instance, fiction documents, research documents, education documents, politics documents and etc. This work does not consider different kinds documents. (ii) The limited coverage of ontology developed on Amharic text news documents. When ontology is developed by two or three individuals its knowledge coverage is limited to the knowledge of those individuals about the entities and relationships. If knowledge base is designed by collaboration of many volunteers around and open for any individual contributors like in Wikipedia it become more semantically reach.

3.7 Summary

Related works on the area indicates that two major approaches can be used for text clustering or classification. The first approach is keyword-based approach and the second is semantic or concepts based approach. Keyword-based approach uses a long list of words as vector space to categorize a given document to a predefined class. This approach is often unsatisfactory for a couple of reasons: first, it keeps the dimensionality of the data very high, and second, it ignores semantics or important relationships between terms like synonyms or antonyms. The two limitations of concept based approach by designing

ontology [9] is: (i) its classification to predefined categories of text documents only which does not consider different kinds documents; (ii) the limited coverage of ontology developed by individuals. Semantic structure (the meaning associated with linguistic units like words) provides access to a large inventory of structured knowledge (the conceptual system). Furthermore, encyclopedic knowledge is grounded in human interaction with others and the world around us that is contributed by any volunteer. The limitation of using only encyclopedic knowledge for feature extraction is the contextual semantics of the document. The meaning of text depends on the aspects of context in which the texts are made. Recently, there is a growing amount of research on how to use encyclopedic knowledge to enhance varied language text mining tasks and also using newly emerging word embedding technology for different text analysis tasks. Although there are many initiatives on text document clustering, there is no work on unsupervised clustering using encyclopedic knowledge with word embedding that improves the performance of text document clustering. Our first contest in text document is the difficulty with identifying significant term features to represent original content by using encyclopedic knowledge. The second contest is related to enriching features using word embedding and reducing data dimensionality without losing essential information's in the text. Thus, our approach considers the contextual semantics using emerging word embedding technology with encyclopedic knowledge from Wikipedia. Thus, our concern is how to design a suitable model for clustering text documents that is capable of improving clustering performance.

.

Chapter 4: Design of Unsupervised Text Document Clustering

4.1 Introduction

In this Chapter, we discuss the proposed design of unsupervised text document clustering using knowledge from encyclopedia with neural word embedding. Furthermore, we focus on the activities of pre-processing of text documents, structuring of encyclopedic knowledge, encyclopedic knowledge based text document representation, neural network based word embedding, text document feature weighting and grouping of related documents (clustering). Text pre-processing activities discussed in this chapter include tokenization, normalization, stop-word removal and stemming. Encyclopedic knowledge consists of structured categorical concept vocabularies and tree like representation of conceptual terms using tree data structure. The activities in neural word embedding technique, word2vec, is also discussed in this Chapter and examples are mentioned in as needed.

4.2 Design Consideration

When we are designing the model, the questions how dimension of large scale text document data is reduced, how representation features are semantically enriched using encyclopedic knowledge with neural word embedding and selecting the appropriate computational measures are technically considered.

a. Dimension Reduction

Dimensionality reduction is defined as a basis of representation within a text document which we can describe most but not all of the variance within our text data, thereby holding the relevant information. Dimension reduction for large-scale text data is attracting much attention nowadays because of increasing amount of text documents and high dimensionality causes serious problem for the efficiency of most of the algorithms [10]. During text document representation, semantic categorical concept features are extracted by mapping with encyclopedic knowledge from online encyclopedia (Wikipedia). The contextual features of these categorical concepts are also extracted using neural network based word embedding. Thus categorical concept features, relationships between concepts and context of concepts are used to represent the text document. This representation handles the semantics of the text and reduces the dimensionality of document vector

representations, so that, it is used to reduce computational overhead, making it easy for real world text document clustering.

b. Feature Enrichment

During representation of documents, features that represent the text documents should be extracted and semantically enriched so that text analytics processes are effective and efficient. The proposed method of using the encyclopedic knowledge with word embedding enriches the features of the text document representation with conceptual terms, concept categories based on relationships from Wikipedia and by contextual terms using neural network based word embedding. Conceptual entities that are related in the form of a tree like structure in Wikipedia are considered during semantic representation of text documents. Using neural word embedding the document vectors are initialized randomly and in the process of training capture the semantics of terms, phrases and sentence. Thus we used encyclopedic knowledge from online encyclopedia (Wikipedia) with neural word embedding enriches text documents semantically. Document features are selected by mapping preprocessed text with Wikipedia and these features are enriched by extracting contextual semantic terms.

c. Selections of Approaches

There are various approaches of text document similarity measures and clustering as we have discussed in Chapter Two. These similarity measures and clustering techniques play a significant role in clustering text documents. Therefore, selection of the most common and appropriate algorithm used for text data clustering is considered. We have selected Spherical K-means clustering that is based on cosine similarity measure.

4.3 Proposed System Architecture

We propose combining encyclopedic knowledge (EK) with the neural network based word embedding to take advantages of the good features both have in semantic based text document clustering. In order to perform text document clustering based on semantic knowledge from Wikipedia with neural word embedding, we considered that the model would have six main components: Structured concept construction, text preprocessing, neural word embedding, text feature extraction, text feature enrichment, and feature weighting and clustering modules. The EK component contains the structured representation of Wikipedia categorical concept vocabularies and tree like relationships

between these categorical concepts. In text preprocessing component, text documents are represented to usable and identifiable format or structure. This component is designed by considering common preprocessing activities and it will be modified depending on language structure. Feature extraction and enrichment component is used to represent a document in a form that it inherently captures semantics of the text. This would help to reduce dimensionality of the text document. Text feature weighting and clustering component assigns numeric value for extracted features that is used to measure similarity or relatedness between text documents during clustering. These components are interconnected in the following Figure 4.1.

Encyclopedic Knowledge
WIKIPEDIA

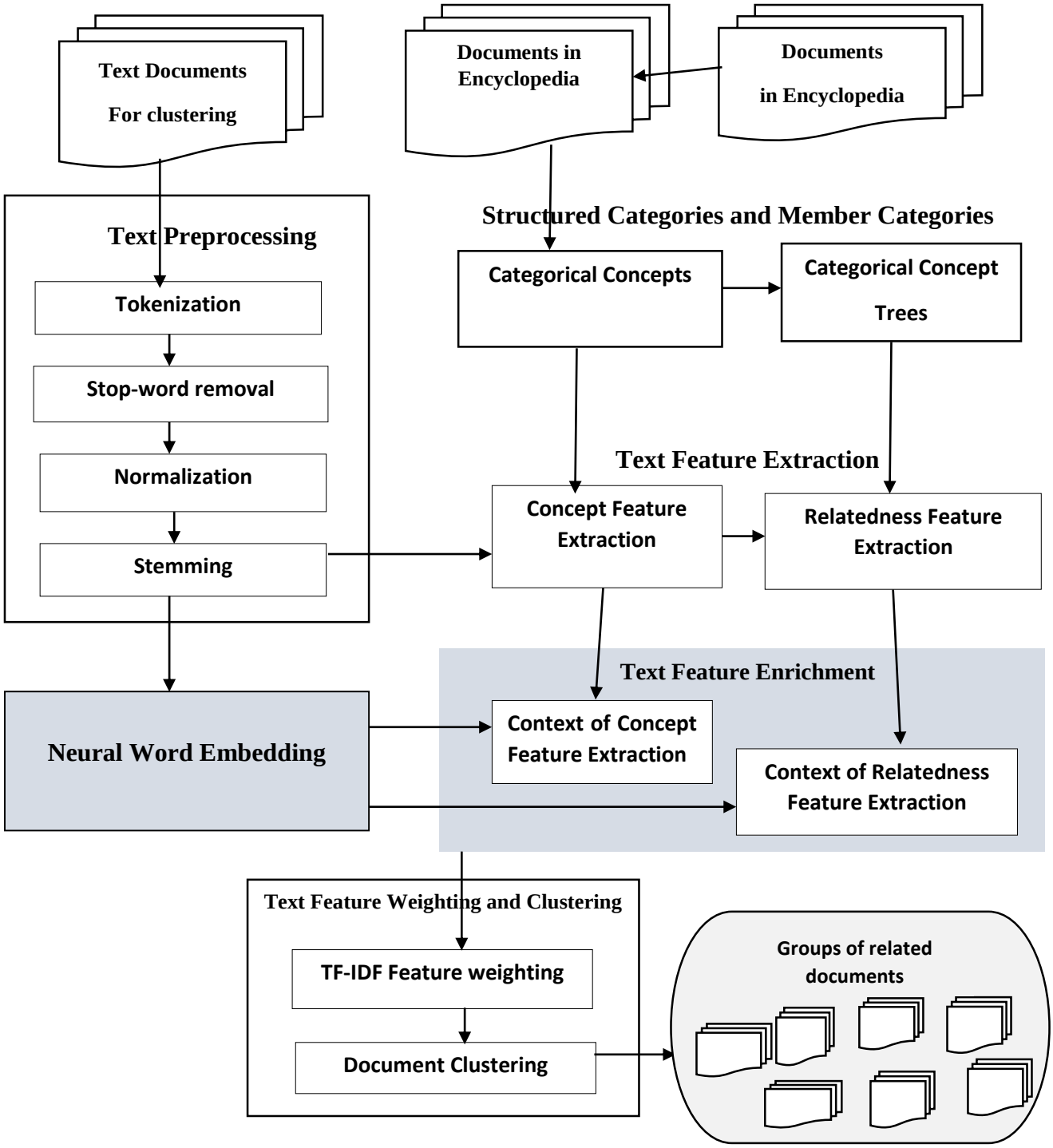


Figure 4.1: *Architecture of Text Document Clustering using EK with Word Embedding*

In next subsections, we describe each of the components of the proposed model in detail using a simple example to elaborate the semantic based clustering ability of our proposed solution showed on Figure 4.1.

4.3.1 Text Preprocessing

The first step in text data analytics process is to create word vector for the text documents to be analyzed but not every word in the text document is important. For instance, text data often contains some special formats like number formats, date formats in language varied texts and the most common words that do not help in text data clustering such as prepositions, articles, and pro-nouns. For this reason, text data must be preprocessed before usage. Preprocessing is about data cleaning or the task that is used to make the text data usable for analysis. The text preprocessing component handles different language specific issues that are imposed by the nature of the language to make the data ready for processing. The preprocessing steps have a huge effect on the success to represent documents in a vector of extracted representative words. The most common pre-processing activities in text data clustering are tokenization, normalization, stop-word removal and stemming.

a. Tokenization

Tokenization is the process of breaking up the given text into units named as tokens or it describes splitting texts into sentences, or sentences into individual words. This is done by locating word boundaries (ending point of a word and beginning of the next word). The tokens may be words, number, punctuation marks, special symbols, etc. After breaking a given text into tokens, data cleaning follows which is the process of removing tokens that have no meaning or that do not change the meaning of the text.

For example, in Amharic a common way to split a text is using certain tokens as a marker, like whitespace or punctuation characters. In the old Amharic writing systems two dots '**:**' ሁለት ነጥብ (*huleti net'ibi*) was used to separate words and now it is replaced by white space. In this work, we used Amharic punctuation marks and white spaces for token identification. It also considers abbreviations and hyphenated words. Forexample, words like ጠ/ሚኒስትር *(t'e/mīnīsitēri),* ቤተ-እስራኤል *(bēte-isira'ēli)* are taken as one word. Tokenization is based upon a set of rules that order to read a sequence of characters as a string and tokenizes them using predefined list of delimiters such as newline, space, dot and hyphen.

b. Stop-word Removal

There are many words in a given text document that are connecting parts of a sentence rather than showing intent of the text. Stop-words are words that occur most frequently in text data, but are not relevant or have no impact to discriminate among text documents.

Because of this they are filtered out before or after processing of natural language data. Stop word removal rules out words with little representative value to the document, e.g., pronouns and punctuations. The common words in English such as ***of***, ***a***, and ***the***, are stop-words and such words are not used to discriminate the text document. For instance, there are common stop words in Amharic which are used for grammatical purposes like ነዉ *(new)*, ነበር *(neber)*, ሆኖም *(honom)*, እና*(ena)*, ነገርግን *(negerigin)*, ሆነ *(hone)*, etc., that are non-informative to identify documents. In order to remove stop-words, a list of stop-words should be identified and listed. We found [20] and also identified a list of such non content bearing words that influence word embedding as listed in Annex A. These stopwords are removed during pre-processing of text documents.

c. Normalization

Various natural language text documents have different features that affect the processing of tokens. Normalization is a process of converting a list of orthographically varied words to a more uniform or common sequence. For instance, in Amharic writing system there are characters with the same pronunciation but different symbols which are named as homophones. The letters such as {አ, ዓ, ዐ}; {ሠ, ሰ}; {ሀ, ኀ, ሐ}; {ጸ, ፀ} are examples of characters which are used interchangeably. This causes, orthographic variants of Amharic texts (for one meaningful element the same word can be written differently in different documents), like ኮምፒዩተር, ኮምፒውተር for *computer*, ማህደር, ማሕደር, ማኅደር for *Maheder* and etc. Amharic texts which refer the same meaningful entity can be written in abbreviation or normal form, example ጠቅላይ ሚኒስትር, ጠ/ሚኒስትር, ጠ. ሚኒስትር for *Prime Ministe*r and መስርያ ቤት, መ/ቤት for *work office,* etc, similarly for hyphenated texts as well. Since there is no standard reference to say that one representation is correct and the other is incorrect, thus characters cause unnecessary increase in the dimension of document representation that causes large data size processing. Then normalization is used to convert such orthographically variant words in to one common representative word before processing of text data. The normalization of Amharic text is done as listed in Annex B.

d. Stemming

In natural language text data processing, stemming is a process where words are reduced to a root by removing inflection through removing unnecessary characters, usually affixes, suffixes and infixes. Stemming is based on the assumption that words with the same root

are referring to the same concept. Conceptually similar words that appear in a text document often have many morphological variations. This is most common in Amharic, which is one of the most morphologically complex languages. For example, stemming will bring the different forms of the Amharic word ኢትዮጵያ (Ethiopia): {የኢትዮጵያ, የኢትዮጵያኖች, ስለኢትዮጵያኖች, በኢትዮጵያኖች, ኢትዮጵያኖችን, የኢትዮጵያኖችን} into their stem word ኢትዮጵያ. This reduces the dimensionality of the Amharic text document during representation of a document which can further improve storage and processing performance. Thus different works are done on developing stemming algorithm for various language texts. For instance, Nega and Willet [46], have developed a stemming algorithm for the Amharic language.

4.3.2 Neural Word Embedding

The relation between word and its meaning is not always unique. There are cases where a word within the same natural language text may have different meanings. In these cases, the assessment of what a text means depends on the aspects of context in which the texts are made. The context of a term in natural language text is given by the interconnection of the different words employed in a sentence for which the semantics of each word is known. As we have discussed in Chapter Two, the techniques have shown their advantages in numerous tasks in natural language processing and effective in finding semantic contexts of texts. Word2vec is neural network based word embedding model that can establish similarities between terms. Using the relatedness between categorical concepts in a given text we can get the most probable contextual words that are not included in text representation using word embedding technique (word2vec).

The following Table 4.1 demonstrates multiple pairs of target and context words as training samples, generated by a 2-word window sliding along the Amharic sentence.

- *“በባህርዳር ከተማ ሲያከናውን የቆየው የአማራ ብሄራዊ ንቅናቄ ፓርቲ በባህርዳር ዩኒቨርስቲ የሚያስተምሩት ዶ/ር ደሳለኝ ጫኔን ፕሬዚዳንት አድርጎ መርጧል።”* – `source BBC/Amharic`
- ***After preprocessing*** - *“ባህርዳር ከተማ ቆየ አማራ ብሄር ንቅናቄ ፓርቲ ባህርዳር ዩኒቨርስቲ ያስተምር ዶ/ር ደሳለኝ ጫኔን ፕሬዚዳንት”.*

Table 4.1*: Pairs of Target and Context Words as Training in Windows Size 2*

Sliding window (size = 2)	Target word	Context
[*ባህርዳር ከተማ ቆየ*]	*ባህርዳር*	*ከተማ ቆየ*
[*ባህርዳር ከተማ ቆየ አማራ*]	ከተማ	*ባህርዳር ቆየ አማራ*
[*"ባህርዳር ከተማ ቆየ አማራ ብሄር*]	*ቆየ*	*ባህርዳር ከተማ አማራ ብሄር*
[*ከተማ ቆየ አማራ ብሄር ንቅናቄ*]	*አማራ*	*ከተማ ቆየ ብሄር ንቅናቄ*
[*ቆየ አማራ ብሄር ንቅናቄ ፓርቲ*]	*ብሄር*	*ቆየ አማራ ንቅናቄ ፓርቲ*
[*አማራ ብሄር ንቅናቄ ፓርቲ ባህርዳር*]	*ንቅናቄ*	*አማራ ብሄር ፓርቲ ባህርዳር*
[*ብሄር ንቅናቄ ፓርቲ ባህርዳር ዩኒቨርስቲ*]	*ፓርቲ*	*ብሄር ንቅናቄ ባህርዳር ዩኒቨርስቲ*

The word vectors are obtained by training neural network on individual words in a text, and given surrounding words as the label to predict the target word or vice-versa. Each context-target pair is treated as a new observation in the data. For example, the target word "*ብሄር*" in the above case produces four training samples: ("*ብሄር*", "*አማራ*"), ("*ብሄር*", "*ንቅናቄ*"), ("*ብሄር*", "*ቆየ*"), and ("*ብሄር*", "*ፓርቲ*"). In this study, we apply the Word2Vec to obtain the fixed-length feature vector, that is, we learn neural network-based word embedding in an unsupervised manner from text.

Algorithm 4.1: *Training Neural Network-based Word Embedding*

```
Start
    Input Text Document Corpus
        preprocess the text ()
          Stem word in texts ()
        add all text in one file F ()
     train F (Word2Vec (F)) ()
     save the trained Model
Stop
Output
      Trained word2vec model feature vector
```

4.3.3 Encyclopedic Knowledge from Wikipedia

Recently, encyclopedic knowledge from Wikipedia has emerged as a useful resource for text analytics tasks. It represents a very large inventory of concepts that are mostly named entities. The basic entries in Wikipedia are articles or links which define the entity, event or a concept to provide conceptual information for the user. Wikipedia contains categorical topic labels that have tree like data structure to each document and these are used as knowledge for different text analytics tasks. Documents of Wikipedia belong to at least one of the categorical concept label and thus concepts are collections of interrelated articles that are represented in phrase or word level. For instance, Table 4.2 shows categorical topics in Amharic Wikipedia represented in the form of word or phrases.

Table 4.2*: Example of conceptual phrases and words from Amharic Wikipedia*

Conceptual labels in Amharic Wikipedia	**Represented as**	**in English**
የኢትዮጵያ ቋንቋዎች (*ye'ītiyop'iya k'wanik'wawochi)*	Phrase	Ethiopian language
ሃይማኖት (*hayimanoti*)	Word	Religion
ፖለቲካ (*poletīka*)	Word	Politics
የኢትዮጵያ ሶማሊያ ጦርነት *(ye'ītiyop'iya somalīya t'orineti)*	Phrase	Ethiopian Somali war
የእንስሳት በሽታዎች *(ye'inisisati beshitawochi)*	Phrase	Animal diseases
የኢትዮጵያ እግር ኳስ *(igiri kwasi)*	Phrase	Ethiopian Football

These categorical concepts are mostly linked entities that are organized by the volunteers. Wikipedia conceptual links can be structured in a manner suitable for further text processing. In our work, the conceptual terms represented in phrase or word level collectively form categorical concept vocabulary represented in its root form.

Categorical Concept Relatedness

Wikipedia consists of conceptual interrelated topic label links in which one follows the other that provide information at each level for the user. We define categorical concept tree to be structural collections of words or phrases for specific conceptual topic label which are menu like paths and Wikipedia links. The tree like relationship in Wikipedia categorical links for a categorical concept $\mathbf{C_i}$, can be represented as:

$C_i \rightarrow \{ C_{i1}, C_{i2}, C_{i3}, C_{i4} \ldots\ldots C_{in} \},$

$C_{i1} \rightarrow \{ C_{i11}, C_{i12}, C_{i13} \ldots\ldots C_{i1n} \}, C_{i2} \rightarrow \{ C_{i21}, C_{i22}, C_{i23} \ldots\ldots C_{i2n} \}, \ldots, C_{in},$

$C_{i11} \rightarrow \{ C_{i111}, C_{i112}, C_{i113} \ldots C_{i11n} \}, C_{i12} \rightarrow \{ C_{i121}, C_{i122}, C_{i123} \ldots C_{i2n} \}, \ldots C_{i1n},$

$C_{i111} \rightarrow \{ C_{i1111}, C_{i1112}, \ldots C_{i111n} \}, C_{i12} \rightarrow \{ C_{i121}, C_{i122}, \ldots C_{i2n} \}, \ldots, C_{i11n}$

$\ldots\ldots\ldots\ldots\ldots, C_{in}.$

Where, $C_{i1}, C_{i2}, C_{i3}, \ldots\ldots, C_{in}.$ represent the related respective links of the categorical concept link C_i.

We use this tree-like structure where in the zero level we have root, in the first level it contains categorical concepts, then related concepts, etc. The main idea of this approach is to keep the number of conceptual relationships per topic label link as much as possible and create a large number of levels that are highly targeted for a specific categorical topic.

For example, Amharic Wikipedia contains the following tree like related links,

- {የኢትዮጵያ_መልከዐ_ምድር→የኢትዮጵያ_ሐይቆች},
- {የኢትዮጵያ_መልከዐ_ምድር→የኢትዮጵያ_ተራሮች},
- {የኢትዮጵያ_ታሪክ→የኢትዮጵያ_ነገሥታት},
- {የኢትዮጵያ_መልከዐ_ምድር→የኢትዮጵያ_ሐይቆች→ዘንገና},
- {የኢትዮጵያ_መልከዐ_ምድር→የኢትዮጵያ_ሐይቆች→ጣና},
- {የኢትዮጵያ_ታሪክ→የኢትዮጵያ_ማዕረግ→ባላምባራስ},
- {የኢትዮጵያ_ታሪክ→የኢትዮጵያ_ማዕረግ→ንጉሠ_ነገሥት_ዘኢትዮጵያ},
- {የኢትዮጵያ_ታሪክ→የኢትዮጵያ_ነገሥታት→አጼ_ቴዎድሮስ}, etc.

Relatedness of these links and other related topic label links from Amharic Wikipedia are shown in the following in Figure 4.2.

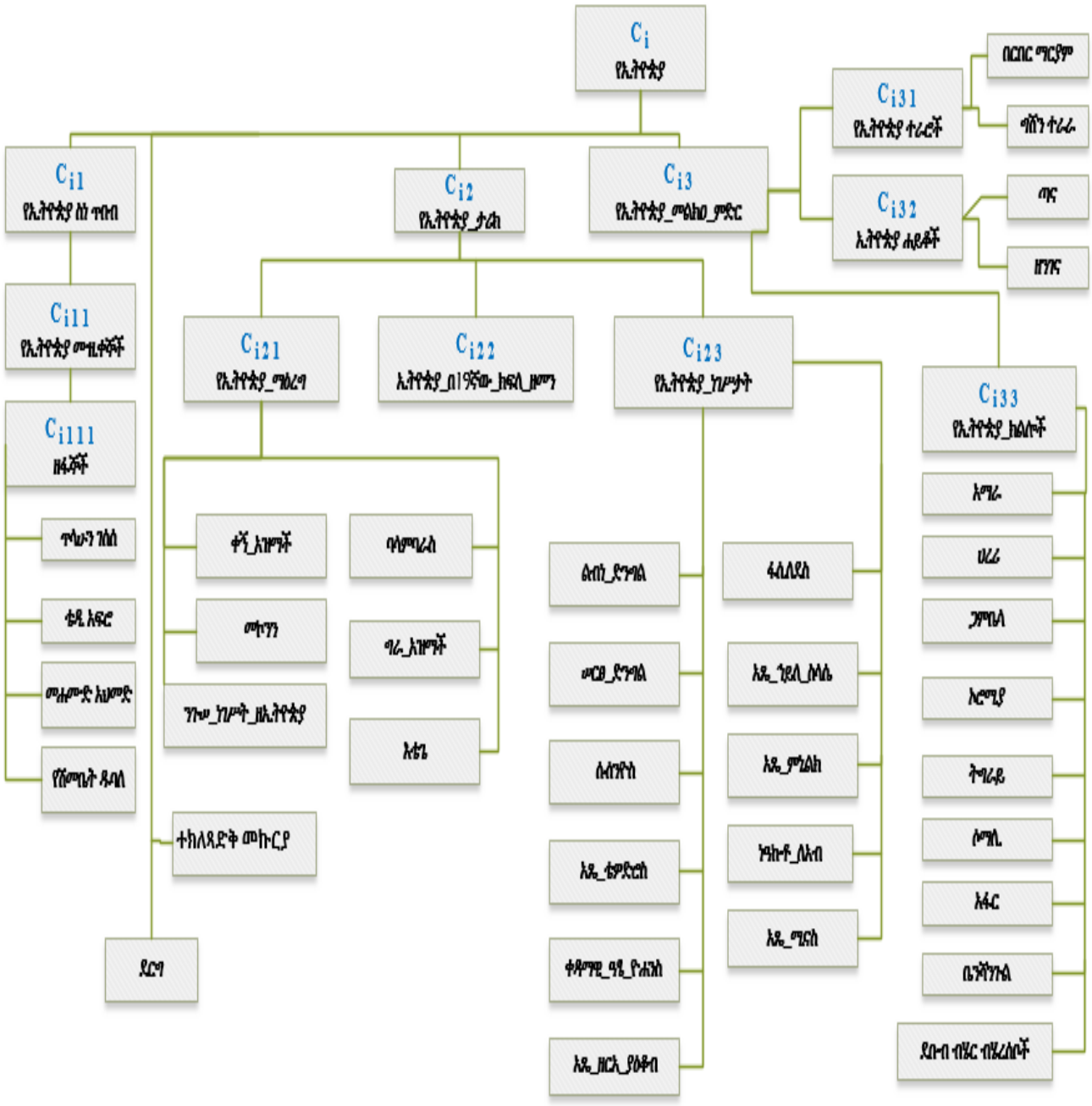


Figure 4.2: *Tree like Conceptual Hierarchies from Amharic Wikipedia*

The relatedness to topic label concept which relate strongly to the node are more likely to be relevant for the category. Furthermore, we define the more related link of Wikipedia concept to be the word or phrase which is an immediate internal or external link of a particular conceptual topic label. For instance, from Amharic Wikipedia{ ኢትዮጵያ_ሥነ-ጽሁፍ → የትንቢት ቀጠሮ }, {ከበደ ሚካኤል → ‘የትንቢት ቀጠሮ}, {የኢትዮጵያ ታሪክ → የኩሽ መንግሥት }, { ናይሎ ሳህራዊ ቋንቋዎች → ጉምዝኛ } are concept labels with the most related link.

4.3.4 Structured Concept Construction

In order to use encyclopedic knowledge from Wikipedia it needs structured representation of the contents depending on the type of text data processing task. We designed structured knowledge representation of Wikipedia conceptual topic labels that can be used for text clustering and other data mining tasks. The proposed structured knowledgebase has catagorical concept vocabularies and linked representation of interrelated categorical vocabularies.

a. Concept Vocabulary Construction

Concept vocabulary is formed from Wikipedia categorical concept links and hierarchical menus that are represented in the form of word or phrase. The categorical topic label links in Wikipedia are concepts that represent entity, a specific action, or thing. Categorical topic label links can be collected and structured in a way that is easy for further text processing. We designed a structure that forms vocabularies of categorical concepts from Wikipedia. In this study, we used Wikipedia database dump that is provided for the users and researchers freely usually twice in a week. Wikipedia backup dumps are available in the form of metadata embedded in XML, in wiki text source and a number of database tables. Database dumps containing categories and categorical links are preprocessed to take out symbols, to identify texts written in other languages and to convert into their root form. The preprocessed Wikipedia texts are stored in a list named categorical concept vocabulary.

b. Categorical Concept Tree Construction

Categorical concept relatedness is structured in the form of tree from Wikipedia categorical topic label successive links and hierarchically related menus. Wikipedia dump consists of categorical concepts with their related links. The relationship between these categorical concepts are preprocessed, structured and represented in tree data structure. Trees are natural, and branching but database tables are man-made rectangles full of numbers and

text. Then to save Wikipedia database dumps in a tree like data structure, in this study we used ltree structure in Postgres database.

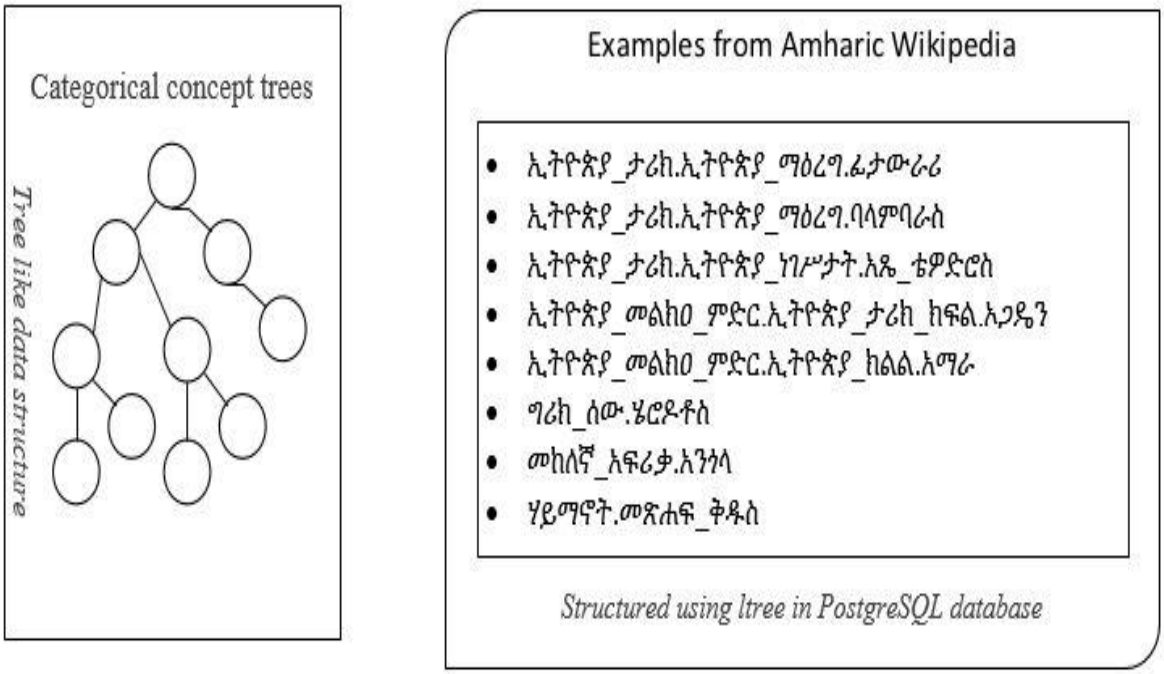


Figure 4.3: *Example of Structuring Categorical Concept Relatedness*

In encyclopedic knowledge structuring process, categorical concept relationships must be preprocessed and represented in tree like structure for further usage as shown in Figure 4.4. In this study we used the ltree extension to save categorical concept tree like relational data in a Postgres table. Ltree allows us to save, query and manipulate tree data structures using a relational database table and implements a materialized path, which is very quick for database operations.

For example, the values in the table shown in Figure 4.4 can be queried as follows to get the parent categorical concept of Amharic "*መጽሀፍ ቅዱስ*" that results *ሃይማኖት*.

SELECT *categorical_concept* ***FROM*** *categorical_relation*
WHERE *parent* ="*መጽሀፍ ቅዱስ*";

Furthermore, by using ltree we can count leaves, cut off branches, and climb up and down trees easily using SQL.

The encyclopedic knowledge from Wikipedia can be structured in a way that is easy for further text document representation process and other data analysis tasks. Using Algorithm 4.1, we structured the encyclopedic knowledge in a database and this is further used for semantic representation of text documents.

Algorithm 4.2: Structuring Categorical Concept Relationships from Wikipedia Dump

```
Start
   Input Wikipedia dump – categorical links
      preprocess the text ()
  repeat
     detect concept and its category from text ()
     If category does not exist
         mark in between texts ()
         Add to concept_relationship ()
     else
         Find category ()
         mark in between texts)
         update concept_relationship ()
   until (categorical links not null)
Stop
Output
     Categorical concept tree (relationship)
```

4.3.5 Text Feature Extraction

The features of each text document are needed to represent a text document for further text mining tasks. Text document feature extraction extracts text information to represent a text message of a document that is the base for different text processing tasks. Text documents are mapped with the structured encyclopedic knowledge to get semantic features of each text document. As we have discussed above, the structured encyclopedic knowledge has a set of categorical concept vocabularies and tree like relations between vocabularies. Documents are mapped to categorical concept vocabularies to create the categorical

concept feature of the text document. These categorical concept features are mapped to structured categorical concept tree representation of the encyclopedic knowledge to create concept relatedness feature of the text document.

a. Concept Feature Extraction

Concept Feature is formed by mapping the preprocessed text documents with categorical concept vocabularies from the structured encyclopedic knowledge. The categorical concept vocabulary consists of the Wikipedia topic label concepts in the form of word or phrase (linked representation of words). This collection of categorical concept vocabulary is mapped against preprocessed text document to extract categorical concepts that shows us the list of conceptual terms within a particular document. Concepts represented in abbreviation or hyphenated form are also extracted by mapping the preprocessed text document with the list abbreviations. In this work, we used the identified lists presented in Annex C. The concept feature of each document is extracted using Equation 9.

$$\boldsymbol{C_{fea} = C_{Co} \cup AH_C} \quad (\mathbf{9})$$

where C_{fea} represents concept feature, C_{co} is the set of mapped categorical concept features, and AH_c is the set of mapped abbreviated or hyphenated concepts.

b. Relatedness Feature Extraction

The structured tree like relationship of Wikipedia link is used to extract the interconnection between conceptual features of text document. Concept-category related feature is formed by mapping extracted conceptual feature of a text document with the structured categorical concept tree of the encyclopedic knowledge. Related feature is a list of related categorical concepts of the concept features. In this study we extracted child and parent of each extracted concepts as the related categorical concepts. The child and parent of a concept is added as feature if the frequency is above the provided threshold. The threshold is given based on the average amount of text on the document. By mapping categorical concept vocabulary, concepts which exist in a document are extracted and used to construct the concept feature of the document. On the other hand, semantic relationship between conceptual terms is extracted using the tree like relationship, in which we named it as related categorical concepts. If the text document contains concept features that are related, then the document will have descriptive related feature. Thus the relatedness features for a document can be extracted using Equation 10.

$$\boldsymbol{R_{fea} = Ch_{Cfea} \cup P_{Cfea}} \quad (10)$$

where R_{fea} represents related feature, Ch_{cfea} is the set of children of concept features, and P_{cfea} is the set of parents of concept feature.

For instance, let Amharic text ***t1*** and ***t2*** contains extracted concept features ***t1*** {ኢትዮጵያ (*ītiyop'iya), ዘፋን(zefani), ቴዲ አፍሮ(tēdī āfiro)}* and ***t2*** *{ስኑ-ጥበብ(sine-t'ibebi), ሙዚቀ(muzīk'e), ጥላሁን ገሰሰ(t'ilahuni gesese)}.* These two text data haven't common concepts. On the other hand the concept features are interrelated to each other. The related features for the two texts are **c*rt1*** {ስነ-ጥበብ, ሙዚቀ}and **c*rt2*** {ዘፋን, ኢትዮጵያ}.

Then the two texts ***t1*** and ***t2*** are represented as:

- ***t1*** = concept features ***ct1*** {ኢትዮጵያ, ዘፋን, ቴዲ አፍሮ} with related **c*rt1*** {ስነ-ጥበብ, ሙዚቀ};
- ***t2*** = concept features ***ct2*** {ስነ-ጥበብ, ሙዚቀ, ጥላሁን ገሰሰ} and ***crt2*** {ዘፋን, ኢትዮጵያ}.

where *ct1*, *ct2* represent the categorical concept feature of text one and text two respectively; and c*rt1*, *crt2* represent categorical related feature of text one and text two respectively.

By using features (***ct1, ct2, crt1, crt2***) the two text documents can be related and compared for further data analysis.

4.3.6 Text Feature Enrichment

The meaning of text depends on the aspects of context in which the texts are used. In practice contextual terms are those that appear frequently in a small number of documents but rarely in the other documents and tend to be more relevant and specific for that particular group of documents, and therefore more useful for finding similar documents. Feature enrichment using context is used to handle the contextual relation of conceptual features. Using word embedding, the proposed solution enhances feature *F* with concepts and relationships from encyclopedic knowledge, which are related to terms in *F*. This process enriches document features contextually.

Algorithm 4.3: *Steps Involved in Context Feature Extraction for a Text Document*

Here Con_{fea}, *Cof*, *RCo*, stands for context feature, concept feature, related categorical concepts (parent (*Pa*) and child's (*Ch*) from the tree) respectively.

Start

Train text documents using *Word2Vec* ()

results, text vector model = (trained_text . mod)

repeat

input ← *Extracted* Cof_i, *RCo*, Doc_i, $Con_{fea}i$← *null*

Con_{fea} ← model.most_similar_2 (Cof_i)

IF (RCoi Contains *Pa* and *Ch*)

Con_{fea} ← Con_{fea} + model.most_similar_2 (Pa + *Ch*)

ELSE Con_{fea} ← Con_{fea} + model.most_similar_2 (RCoi)

IF (Doc_i Contains Con_{fea})

$Con_{fea}i$ ← $Con_{fea}i$ + Con_{fea}

until *(text document* $(Con_{fea}n)$*! = null*)

Stop

output

$Con_{fea}i$ (Context Feature of a document)

By mapping tree like concept – category relationship, related concepts of a text document are extracted and added to a feature of a document. On the other hand, the context of each related feature is extracted and added using Algorithm 4.3. Given the related features, in what context the relationship is handled. The context of each concept feature is also extracted and used based on the frequency threshold. Then the context feature can be represented as:

$$\boldsymbol{Con_{fea} = Con_{Cfea} \cup Con_{Rfea}} \quad (11)$$

where Con_{fea} represents context feature, Con_{cfea} is the set of contexts of concept features, and Con_{Rfea} is the set of contexts of related feature. Then the descriptive contexts are added to the document feature.

For instance, let Amharic text ***t1, t2, t3*** contains concept-category related features ***t1****{አማራ → ክልል, ጠና }*, ***t2***{ሙዚቃ, ዘፋን→ቴዲ አፍሮ} and ***t3***{አባት, እናት →ቤተሰብ}.

Then we can think of the following examples of semantic relations:

- [ቴዲ አፍሮ] + [ዘፋን] ~= [አቦጊዳ, ጥቁር_ሰው],
- [ቤተሰብ] - [እናት] + [አባት] ~= [ልጅ],
- [*አማራ*] + [*ክልል*] + [*ጠና*] ~= [ገዳም, አሳ].
- [*አማራ*] + [*ክልል*] ~= [*ጠና*]

Furthermore, the contexts of the semantically related concept features are probably quite similar or related in meaning. These contextual features are extracted using the newly emerging technology named as word embedding.

In this study, we applied word embedding technique word2vec to obtain the fixed-length feature vector, in which we get learning result of neural network-based word embedding in an unsupervised manner from text documents. The vector representations of words learned by Word2Vec model has been shown to carry semantic meanings. Based on the learning (word2vec) result vector of collection of texts, we define contexts of a related categorical concept in a text as terms obtained by the interconnection of the different conceptual terms employed in a text. The interconnections between concept features in the document are used to find the most probable contextual term of the text document. We used the relatedness of categorical concepts in encyclopedic knowledge, to find the conceptual terms that have very similar neighbors in which these terms are probably quite similar or related in meaning. Text features are extracted for each text document using Algorithm 4.4. The importance of the feature is evaluated using the text weighting process. These features with lowest weight (uncorrelated) are eliminated to make more weighted concepts more descriptive.

Algorithm 4.4: *Generic steps involved in enriched text document feature extraction*

Here td_f, *Cof*, *AHc*, *RCo*, *ConCo* stands for text document feature, categorical concept feature, abbreviated and hyphenated concepts, related categorical concepts (parent (*Pa*) and child's (*Ch*) from the tree) and contexts of concepts respectively.

Start

Preprocess all text documents ()

Train all preprocessed text documents using *Word2Vec* ()

results, text vector model (trained_text . mod)

repeat

input ← *preprocessed* doc_i , td_f ← *null*

AHc ← detect conceptual abbreviations(), $doc_i = doc_i - AHc$

td_f ← td_f + *AHc*

Cof ← detect concept features ()

td_f ← td_f + *Cof*

RCo or *pa,ch* ← find related categorical concepts given (td_f)

td_f ← td_f + *pa* +*ch*

ConCo ← *find Context feature, given* trained_text . mod

td_f ← td_f + *ConCo*

until *(text document* (doc_n)*!= null)*

Stop

output

td_f (features of text document)

4.3.7 Text Feature Weighting and Clustering

In previous section 4.3.6 we described that representative features of a document are constructed by extracting ***n*** number of conceptual terms with ***r*** number of their most related categorical terms using encyclopedic knowledge and ***cr*** number of context based related categorical terms. Once the semantically representative terms are extracted from the text document, the documents have to be transformed in to a document vector or something the clustering algorithm can work with. As discussed in section 2.5.2 there are various methods

of text representation. In this work we used vector space model TF-IDF weighting scheme which is the most common representation technique used in different text analytics and it gives us how important is a conceptual term to a document in a collection, since it takes in consideration not only the isolated term but also the term within the document collection.

a. TF-IDF Weighting

Using vector space model TF-IDF, text documents are represented as vectors in an n-dimensional space, where n is the number of conceptual terms or concept-related terms. The weight value of each term is computed by term frequency and inverse document frequency.

- ***TF(ct)*** = (Number of times categorical term *ct* appears in a document) / (Total number of categorical terms in the document).:
- ***IDF(ct)*** = **log** (Total number of documents / Number of documents with term ***t***).

Then, for a conceptual term ***t*** in a document ***d***, the weight **W*t,d*** of term ***t*** in document ***d*** is given by computing the following:

$$\boldsymbol{Wt,d} = \boldsymbol{TFt,d} * \boldsymbol{log}\left(\frac{\boldsymbol{N}}{\boldsymbol{DFt}}\right) \quad (12)$$

where, TF*t,d* is the number of occurrences of *t* in document *d* and D*Ft* is the number of documents containing the term *t*.

Therefore, tf-idf scheme assigns to a conceptual term ***t*** a weight in document ***d*** that is:

- Highest when ***t*** occurs many times within a small number of documents.
- Lower when the term occurs in smaller number of times in a document, or occurs in many documents.
- Lowest when the term exists in all documents.

Furthermore, when we consider a text collection consisting of 1000 documents, a term which appears in each of the documents in the text collection is almost useless with lowest weight; Since it does not provide information for further data analysis (clustering).

Thus text document doc *i* can be described as [Wi_1, Wi_2, Wi_3,.....Wi_j,....Wi_n], where Wi_j is weight value of j the term in the n-dimensional vector space.

b. Text Document Clustering

K-means clustering is the most commonly used unsupervised machine learning algorithm for partitioning a given data. There are several k-means algorithms available. In this study, we used spherical k-means algorithm for clustering text documents. The spherical k-means algorithm, i.e., the k-means algorithm with cosine similarity, is a popular method for clustering high-dimensional text data. In this algorithm, each document as well as each cluster mean is represented as a high-dimensional unit-length vector. We used the most popular similarity measure, i.e., cosine similarity, which measures the angle between the document vectors. Unfortunately, there is no global theoretical solution to find the optimal number of clusters for any given data. A simple and common approach is to compare the results of multiple executions with different k classes and choose the best one according to a given characteristics of the data. To evaluate unsupervised text document clustering using encyclopedic knowledge with neural word embedding we used a given K classes of documents.

4.4 Summary

In this chapter design of unsupervised text document clustering using encyclopedic knowledge with neural word embedding was presented. We described the basic design criteria and elements of the model for text document clustering. The model consists of document preprocessing module (tokenization, normalization, stop-word removal and stemming) that is essential for better extraction. Structuring encyclopedic knowledge module consists of representations of categorical concepts and the tree like linked representation of these concepts. Selection of text feature item is a basic and important matter for text mining and information retrieval. Feature extraction extracts representative features relevant to the original text sets, so as to reduce the dimensionality of feature vector spaces. Mapping module includes relating text documents with encyclopedic knowledge and finding the representative features of the text (categorical concepts, related categorical concepts). Neural word embedding vector representation of text document captures the distributional representations of words. The contextual terms of the related categorical concepts are extracted using word2vec text vector. During this study, out of different alternatives, the selected methodology has been discussed. The approaches that we described illustrates how the knowledge is structured from encyclopedic knowledge and how it is used with neural word embedding technology.

Chapter 5: Experimentation and Evaluation

5.1 Introduction

This Chapter describes the experimentation activities, procedures and evaluation of the results using evaluation methods. Mainly the natures of text data collected, the text features extraction, weighting features and the clustering results of the experiment are discussed in this chapter. The comparative analysis of the clustering results is discussed. The snapshots of the experimentation procedures and evaluations of the results are explained.

5.2 Experimental Procedures

The following subsections discuss the activities and steps to evaluate unsupervised text document clustering using Encyclopedic knowledge from Wikipedia and neural word embedding for selecting semantic representative features.

5.2.1 Data Collection

The experimentation of unsupervised text document clustering begins with collection of text documents from different well-known data stores. We have collected two types of Amharic data for experimentation. (1) Amharic Wikipedia database dump that is structured and used as encyclopedic knowledge base, and (2) Amharic text document corpuses that are collected from different categories of documents for experimentation.

a. Amharic Wikipedia Data

A complete copy of all Wikimedia links, concepts, category-links, in the form of wikitext source and a number of raw database tables are provided for users usually twice a month. We have collected Amharic Wikipedia database dump available on the date June 3, 2018 consists of category and categorical link tables. The table consists of 29,042 row of data with its columns categorical concepts and related links. We used PostgreSQL database that is object-relational database management system to store and structure Amharic database dump. PostgreSQL provides extension ***ltree*** data type for representing and processing labels of data stored in tree-like structure. By using ltree we structured the preprocessed related conceptual links of Wikipedia data.

b. Amharic Text Documents

Amharic text document data are collected from Amharic bible, news agencies, broadcasting media, online newspapers and magazines. We use different sources to make data heterogeneous and writer independent. Out of many text documents available, 3885 are randomly selected for testing. Based on the text contents are discussing about, these documents are categorized manually by domain experts.

Table 5.1: *Manually categorized text documents collected from different sources*

No	Class Name	Number of Text Documents Collected	Sources of Data Collection	Doc. Code
1	Politics	760	Ethiopian-reporter, BBC Amharic	Pol
2	Religious	1060	Amharic Bible	Rel
3	Technology	815	Fana BC, Ethiopian-reporter, ENA	Tec
4	Business	370	Ethiopian-reporter, EBC	Bus
5	Health	200	TenaAdam, mehedratena	Hel
6	Art	230	Ethiopian-reporter, Addis Admass	Art
7	Sport	450	Ethiopian-reporter, Fana BC	Spo
Total		**3,885**		

As shown in Table 5.1, we have collected 3,885 text documents from 7 different categories. These groups of text documents are manually assigned category name by the experts and the writers of the documents. For example, 1118 text documents collected from the whole Amharic bible, i.e., old testament (OT) and new testament (NT) are categorized into religious text documents; 1002 text documents collected from Ethiopian-reporter newspaper are categorized into politics by the writers and random samples are checked by the expert. These manually categorized text documents are used as for checking the results of unsupervised text documents clustering using encyclopedic knowledge with neural word embedding.

5.2.2 Tools and Programming Languages

a. Tools

In this experimentation, we have used Anaconda and Netbeans IDE programming tools. Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications. We have trained text document corpus using Gensim which is a robust open-source vector space modeling and topic modeling toolkit implemented in Python. In Gensim a corpus is simply an object, when iterated over, returns its documents represented as sparse vectors. Netbeans IDE is a free and open source integrated development environment for application development on Windows, and other operating systems. The IDE simplifies the development of web, enterprise, desktop, and different applications that use the Java and HTML5 platforms. The IDE also offers support for the development of PHP and C/C++ applications.

b. Programming Languages

Python is a powerful high-level, object-oriented programming general-purpose language. It has wide range of applications from Web, scientific and mathematical computing (SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). In this experimentation process we have used Python for machine learning tasks and Netbeans for all other steps.

c. Experimental Setting

In this study we used a computer with a memory capacity of 8 GB RAM, 2.2 GHZ processor and 64 bit windows 10 operating system.

5.2.3 Text Data Cleaning

After data collection, the next step in the experimentation of text data clustering is text data cleaning (preprocessing) and creating document vector representations. The preprocessing tasks explained in Chapter Four are adopted on the whole collected Amharic text data. Then we have collected Amharic stop-words from previous works and manually identified a total of 900 stop-words (words that do not change the meaning of the document) as listed in Annex A. We applied stop-word removal for texts that are used for testing and training neural word embedding. Finally, in text preprocessing step, we have used Amharic stemming algorithm developed by Alemayehu, Nega, and Willett [46].

5.2.4 Neural Word Embedding

The preprocessed text documents are trained using neural word embedding technique. We applied Word2Vec approach is for representing a word based on its embedding. Word2Vec generates a set of vectors, one vector for each word found in the text corpus. We considered a fixed four size word window that slide across the text and feature learning approach. Skip-gram model is trained that is used further to predict the surrounding words in the window given the current categorical concepts. We have trained two times 8,658 number of text documents with more than 1.5 million words and resulted trained vector model. The output has dimension 1x*V*, where *V* is the vocabulary size, that represent one-hot encoding of a word. Based on this word embedding vector result, the relational operations between words like distance and analogy is made. Table 5.3 shows the examples of the distance between terms in trained model.

Table 5.2*: Instances of the Distance Values Within Words in Embedding Vector*

Terms	**Word Embedding Distance -Training Result**					
ህመም	መደንዘዝ	ትውከት	ኩላሊት	ድካም	እጢ	ስቃይ
	0.49915186	0.4976017	0.493521	0.49348527	0.47665387	0.4699025
ፊዴራል	ፖሊስ	ማረሚያ	አዲተር	መስሪያ	አስተዳድር	ኮሚሽነር
	0.6342526	0.47735998	0.4070405	0.39875942	0.3750555	0.3718779
ሞባይል	አፕሊኬሽን	ኤስ	ኢትዮ	ስማርት	ኦፖሬቲንግ	ኦፖሬቲንግ
	0.46558622	0.46496454	0.4630726	0.45355147	0.44530717	0.44530717
ዶላር	ገንዘብ	ብር	ሚሊዮን	ሺህ	አይቲኤፍ	አሜሪካ
	0.5998056	0.53900427	0.53072107	0.5099767	0.4909553	0.43341544
ኮምፒውተር	ኤችፒ	ራንሰም	ላፕቶፕ	ኪቦርድ	ዴስክቶፕ	ማክ
	0.6109342	0.5578999	0.5538677	0.5536411	0.54935753	0.5413407
ዋንጫ	ቻን	ማጣሪያ	ሻምፒዮን	ሴካፍ	አፍሪካ	ተሸንፎ
	0.5628299	0.5379731	0.5280415	0.49070022	0.47065616	0.48836362
ኢህአዴግ	ዴሞክራሲ	አጀንዳ	ብአዴን	ፓርቲ	አብዮት	ድርጅት
	0.55041987	0.5483947	0.5295941	0.51064366	0.50294995	0.48187944

From Table 5.3, the Euclidian distance between the first six most related contextual terms was listed with the values from the trained model. In order to use insight from the trained model, word distance and analogy are used during feature enrichment process.

5.2.5 Feature Extraction, Enrichment and Weighting

The next activity is generating representative features of preprocessed text documents by mapping with the encyclopedic knowledge and enriching the feature representation using word embedding results. We followed the following steps to extract and enrich features from preprocessed text documents.

1) Mapping with abbreviated and hyphenated concepts.
2) Mapping with categorical concept vocabularies.
3) Mapping with tree like relationships of categorical concepts.
4) Finding the contexts using trained model.
5) Assign the features for the given document, the detail is explained in section 4.3.5.

By using these steps, the features are generated for each text document. The extracted features are represented and weighted by using commonly used text weighting method vector space model TF-IDF. The weight value of each conceptual term in the document are computed by term frequency and inverse document frequency.

5.2.6 Applying Spherical k-means

The final activity is grouping of related documents 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. Figure 5.1 shows the sample result obtained that represents a document with Document-ID and its clustered group id Cluster-ID and more sample outputs are presented in Annex D.

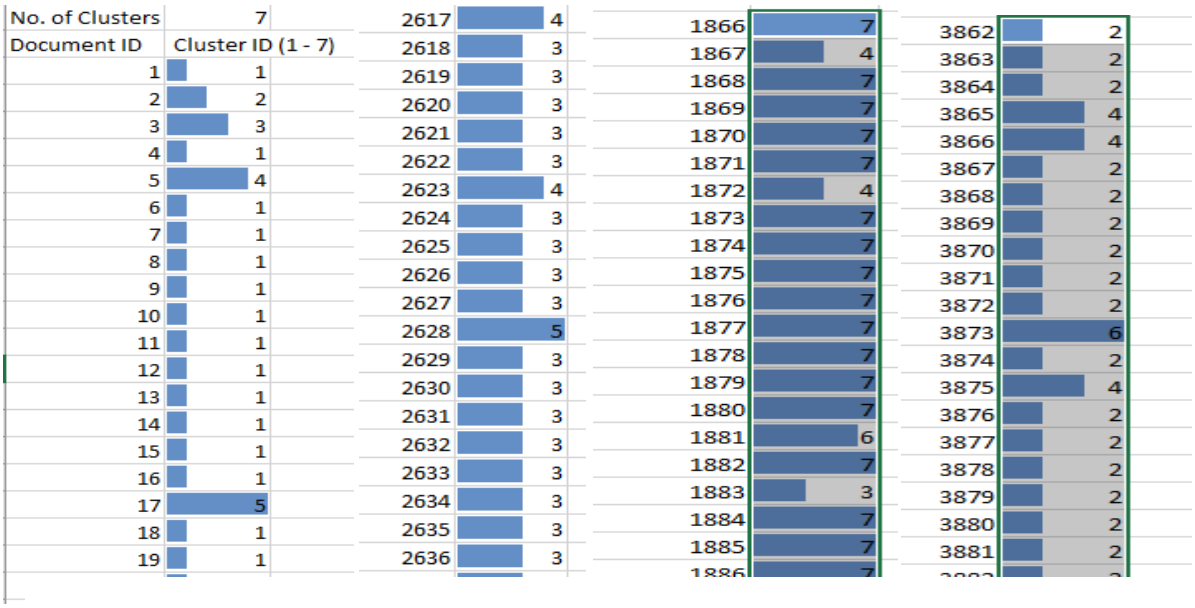


Figure 5.1: *Sample Snapshot of Clustering Result*

5.3 Evaluation

To evaluate the clustering results, the comparison was done between the document clustered using unsupervised method with that of manually grouped by experts. Precision and recall were calculated. These measures try to estimate whether the prediction was correct with respect to the underlying true categories.

To compare our result with the results of the most related work done on text document clustering using Wikipedia, evaluation of clustering was not mentioned. Thus, we have also done experimentation by skipping feature enrichment processes (finding and mapping contexts of related concepts, contexts of concept features using word embedding) for further analysis and result comparisons, i.e., we have tested text document clustering by only using the encyclopedic knowledge for feature extraction. The confusion matrix, precision, recall and accuracy are used for evaluation.

5.4.1 Confusion Matrix

Confusion Matrix for Clustering using EK with Word Embedding

Table 5.3: *Confusion matrix for clustering using EK with word embedding results*

Actual class / clustered number of documents	Politics	Technology	Business	Health	Art	Sport	Religious	**Total**
Politics	691	5	14	4	2	30	13	**759**
Technology	7	747	8	17	21	14	1	**815**
Business	4	5	333	0	0	26	2	**370**
Health	0	0	2	182	2	9	5	**200**
Art	3	0	0	1	214	10	2	**230**
Sport	3	3	6	3	0	435	0	**450**
Religious	21	0	0	1	0	0	1036	**1058**
Total	**729**	**760**	**363**	**208**	**239**	**524**	**1059**	**3,882**

Based on these results, the overall error rate (the frequency of clustering that will predict to wrong class) is **0.066**. The value of f-measure (measurement that represents both precision and recall) is **0.964**.

The above distribution of text documents into different category shows that different classes have different conceptual related terms. For instance, the document clustering of technology class test document distributed 21 in art class, 17 in health class, 14 in sport

class, 8 in business class, 7 in politics class and 1 in religious class. Here the amount of distributed documents indicates that art class documents will have more amount of conceptually related terms with technology category than other categories. The distribution also indicates the number of text documents that are incorrectly clustered.

Confusion Matrix for Clustering using only EK

Table 5.4: *Confusion matrix for clustering using only EK results*

Actual class clustered number of documents	Politics	Technology	Business	Health	Art	Sport	Religious	**Total**
Politics	662	3	11	7	1	64	11	**759**
Technology	10	705	21	28	18	21	2	**815**
Business	9	8	295	0	0	55	3	**370**
Health	1	0	2	189	1	4	3	**200**
Art	2	0	0	58	158	11	1	**230**
Sport	3	1	11	9	2	422	2	**450**
Religious	18	0	0	1	0	0	1039	**1058**
Total	**705**	**719**	**340**	**292**	**180**	**577**	**1061**	**3,882**

Based on these results, the overall error rate (the frequency of clustering that will predict to wrong class) is **0.106**. The value of f-measure (measurement that represents both precision and recall) is **0.939**.

The relatedness between clusters as seen in the can be described as the more text documents grouped incorrectly to specific class denotes the two clusters are more conceptually related. This represents that the two clusters have more conceptual terms or their relationships that allows incorrect clustering of these types of text documents.

5.4.2 Precision, Recall and Accuracy

The result of clustering is evaluated using metrics precision (P), recall (R) and accuracy (A) in percentage (%) for both experimentations and the detail result is provided in Table 5.4 below.

Table 5.5*: Evaluations of clustering using EK with and without word embedding*

Class Name	*No of Input Documents*	*With Word Embedding*			*Without Word Embedding*		
		P	***R***	**A In %**	***P***	***R***	**A In %**
Religious	1060	0.98	0.99	98.99	0.98	0.99	98.99
Politics	760	0.91	0.99	91.91	0.87	0.99	87.88
Technology	815	0.91	1.0	91.00	0.87	1.00	87.00
Business	370	0.90	1.0	90.00	0.80	1.00	80.00
Health	200	0.91	1.0	91.00	0.95	1.00	95.00
Art	230	0.93	1.0	93.00	0.69	1.00	69.00
Sport	450	0.97	1.0	97.00	0.94	1.00	94.00
Total	**3,885**	**0.94**	**0.99**	**94.95**	**0.894**	**0.99**	**90.29**

The measures in above Table 5.4 shows whether the prediction of each text documents class as being in the same cluster was correct with respect to the underlying true categories. It shows the result of clustering text documents using only encyclopedic knowledge and using encyclopedic knowledge with word embedding for feature enrichment.

For instance, the clustering accuracy of business category documents using Encyclopedic Knowledge (EK) with word embedding is 90.00 %. Out of 3,885, the total number of correctly clustered text documents using only EK are 3470 which is 90.29% of the total. Figure 5.9 shows the comparisons of accuracy values of the two test results of clustering for each class of text documents.

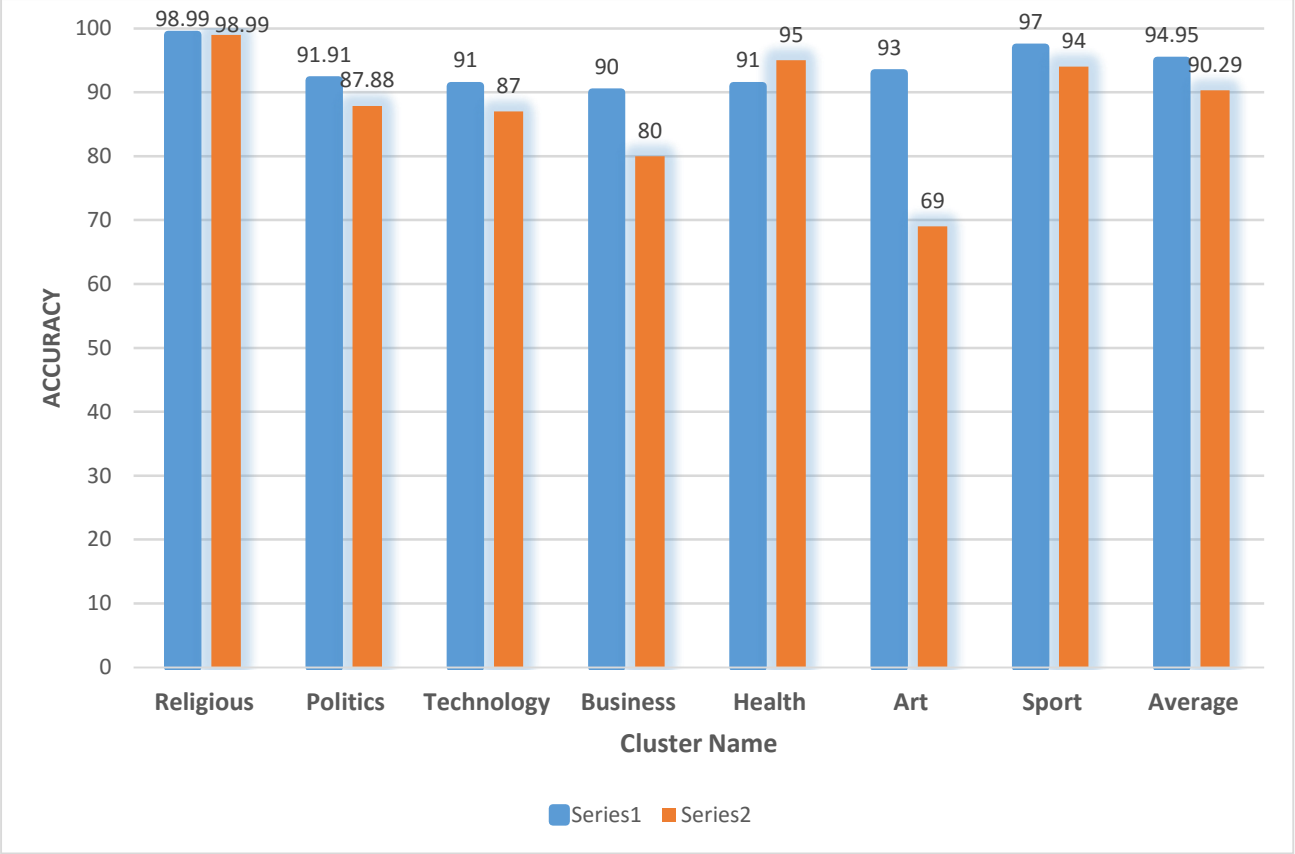


Figure 5.2: *Accuracy value Difference of Clustering with and without WE*

The Figure 5.9 shows the accuracy values of each cluster. The series1 shows the accuracy result of clustering text documents using encyclopedic knowledge with word embedding technology for feature extraction and series2 shows the accuracy results of text clustering using only EK for feature extraction. The difference between accuracy values between text clustering using EK with and without word embedding is clearly seen in Figure 5.9.

5.4.3 Average Accuracy Values Vs Cluster Size

For testing our proposed system, we used the cluster size K value 7 for seven types of text documents grouped by the experts. To see the effect of cluster size on the average accuracy value, we have tested by using 4, 5, and 7 kinds of documents from the collected corpus. Thus, by using cluster size K value 4, 5 and 7, in Table 5.7 are obtained and evaluated in Table 5.8.

Table 5.6: *Clustering Result using Different Cluster Size (K).*

No	*Class Name*	*No of Input Documents*	*Without Word Embedding*			*With Word Embedding*		
			TC K=4	*TC K=5*	*TC K=7*	*TC K=4*	*TC K=5*	*TC K=7*
1.	Religious	1060	1057	1057	1039	1052	1052	1036
2.	Politics	760	**NI**	**NI**	662	**NI**	757	691
3.	Technology	815	**NI**	400	705	**NI**	**NI**	747
4.	Business	370	365	322	295	368	363	333
5.	Health	200	198	198	189	193	184	182
6.	Art	230	226	150	158	156	222	214
7.	Sport	450	**NI**	**NI**	422	**NI**	**NI**	435

In Tables 5.7 and 5.8, **NI** denotes not included during clustering, K represents the cluster size and TC represents the number of documents which are clustered correctly.

Table 5.7: *Accuracy Values of Clustering using Different Cluster Size (K).*

No	*Class Name*	*No of Input Documents*	*Without Word Embedding*			*With Word Embedding*		
			A (%) K=4	*A (%) K=5*	*A (%) K=7*	*A (%) K=4*	*A (%) K=5*	*A (%) K=7*
1.	Religious	1060	98.99	98.99	98.99	99.24	99.24	98.99
2.	Politics	760	**NI**	**NI**	87.88	**NI**	99.75	91.91
3.	Technology	815	**NI**	50.01	87.00	**NI**	**NI**	91.00
4.	Business	370	98.65	87.03	80.00	99.46	98.11	90.00
5.	Health	200	99.00	99.00	95.00	96.50	92.00	91.00
6.	Art	230	98.26	65.27	69.00	67.83	96.52	93.00
7.	Sport	450	**NI**	**NI**	94.00	**NI**	**NI**	97.00
Average			**98.73**	**80.06**	**90.29**	**90.78**	**97.12**	**94.95**

Direction of Average Accuracy Values

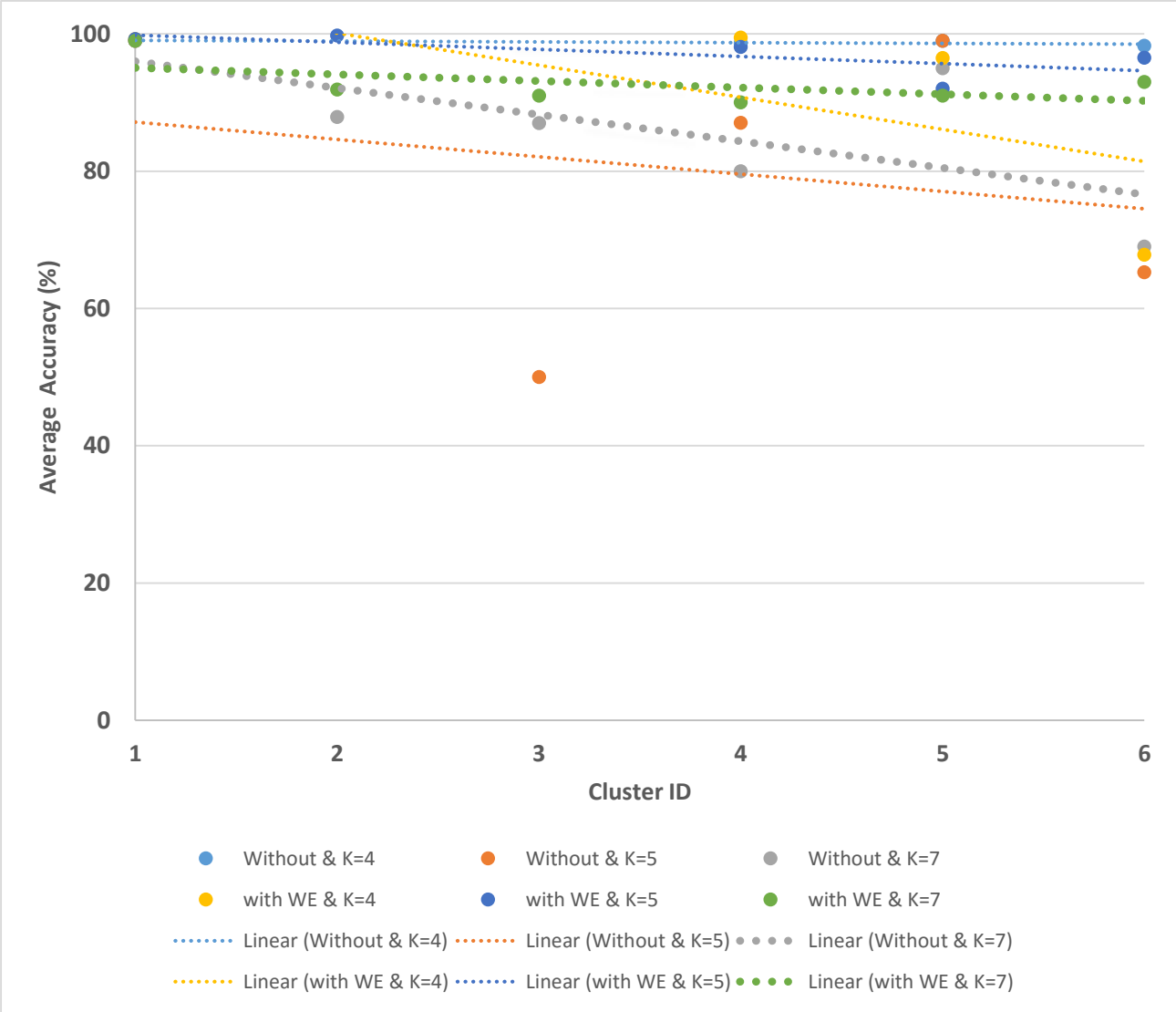


Figure 5.3*: Direction of Average Clustering Accuracy Vs Cluster Size*

Figure 5.11 shows lines on a graph representing the general direction that a group of accuracy value points with different cluster size seem to be heading. In this table WE denotes with word embedding, without denotes without using word embedding and K represents the size of cluster. The cluster id 1, 2, 3, 4, 5 and 6 represents the cluster name religious, politics, technology, business, health and art respectively. The linear series shows the direction of Average accuracy of text document clustering by changing the size of cluster to 4, 5 and 7. The directions of Accuracy values are showen in both test cases, i.e., text documents clustering using encyclopedic knowledge with word embedding (WE) technology and clustering text documents using only encyclopedic knowledge for feature

extraction. The linear trendline (linear series) shows the increasing or decreasing rate of accuracy that is a best-fit line to compare the two clustering results with change in the cluster size (K). Here we notice that changing the size of the class has a significant effect on the rate of accuracy. The difference of the two best-fit lines for cluster size 4, 5, and 7 are shown clearly in Figure 5.11. The gap between lines using K=7 (linear (with WE & K=7), linear (without & K=7)) shows the performance improvement of using EK with word embedding in which the rate of accuracy decrease significantly with increase in cluster size. The detail result analysis is discussed on the section 5.4.

5.4 Discussions

Out of 3,885 documents used for testing, some documents are clustered incorrectly, some are missed and others are clustered correctly. Three documents (one from political category and two from religious category) are missed in clustering process. We have analyzed that these documents contain short texts and low representative conceptual term found during feature extraction process. During weighting process also, we have used threshold value to enrich high weighted terms and to eliminate low representative conceptual terms. Because of these reasons three documents are missed during clustering process.

Totally 3470 text documents are clustered correctly to their respective categories. Documents of sport and religious category are more accurately clustered because of less conceptual relations with other categories. As shown in distribution charts, the documents in religious category are not distributed wrongly into sport, business, art and technology which shows low conceptual relationship between these categories. The text documents in technology category and politics are distributed in all classes during clustering in which the amount of distribution varies. The distribution of these documents show that politics and technology classes have common or related conceptual terms that has the effect in short text document clustering. There are more conceptual terms used in politics and technology that can be probably used in all categories of documents. These and other document distribution differences during clustering come due to the conceptual relationship between categories.

During this study, we have tested similar test documents using encyclopedic knowledge with word embedding and without word embedding in feature extraction process. As shown in Table 5.6, the result of religious text document clustering shows that 3 more text

documents are clustered correctly without using word embedding. Moreover, 7 more documents of health category are clustered correctly without using word embedding. On the other hand, 84 more text documents i.e. 2.16% of the total are added incorrectly into health class without using word embedding. This shows that adding contextual terms for religious and health documents would predict to wrong group that would be adapted if more documents are added. But when we compare all other clustering results, feature enrichment using encyclopedic knowledge with word embedding adds total 168 more text documents to their correct category. This shows that using encyclopedic knowledge with word embedding (our proposed system) for feature enrichment clusters with an average increment of 4.32 % than that of using only encyclopedic knowledge. The result of testing is shown Table 5.6. The change in accuracy of the two clustering results is shown clearly in Figure 5.11 by using best-fit line of the average accuracy values. Furthermore, when we compare the related work done using similar natural language (Amharic) text documents, i.e., concept based Amharic text categorization discussed in Section 3.6, it shows significant increment in performance even though it was classification, based on different data and approaches.

During this study, we have tried to see the relationships between cluster size and average accuracy values for each class of documents. The small difference between average accuracy values of each class with clustering using cluster size K (4, 5 and 7) is shown in Table 5.7. In both test cases, the best-fit lines in Figure 5.11 show that changing the size of the cluster has a significant effect on the rate of clustering accuracy. Changing rate of clustering accuracy of our proposed system is smaller as compared with using only encyclopedic knowledge when cluster size increases. The factors are the conceptual or semantic relationships between the clusters. Furthermore, these best-fit lines vary significantly that conveys using encyclopedic knowledge with word embedding for text document clustering can improve accuracy.

There are 254 wrongly clustered text documents. Even though these documents describe about issues out of assigned cluster, the conceptual terms in the documents are more close to incorrect cluster representative. Therefore, these documents were grouped under incorrect cluster. There are also language problems we got during cluster analysis that will probably lead to wrong categorization of documents. Under preprocessing module there are tokenizer, stop word-removal, and stemmer in which we are used including their own

linguistic limitations during testing. The documents are tokenized based on the space in between terms in the text but we don't get all list of phrasal terms in Amharic that cannot be tokenized using space. In Amharic there is no list that contains all stop-words. We have used Amharic stemmer that had accuracy of 95% as discussed in section 4.3.1 of chapter four. These linguistic problems listed above had their own implication during feature extraction and weighting that have influence on wrongly clustered text documents. In addition to the above reasons, the limitations on the enrichment of Amharic Wikipedia that was available recently had its own implication because the result primarily depends on the represented encyclopedic knowledge.

Therefore, having enriched encyclopedic knowledge and capability of linguistic preprocessing tools decides the final result of clustering text documents. This shows that as the encyclopedic knowledge base gets richer, the performance of the system will be improved significantly. During this study, for seven clusters with a total of 3,885 text documents, the result of correctly clustered document found from the experiment is 94.95%, which shows the average accuracy of the clustering is good. These testing results validate the effectiveness of the proposed system.

Chapter 6 : Conclusion and Future Works

6.1 Conclusion

This research work had attempted to look into the techniques of unsupervised text document clustering by enriching text features using encyclopedic knowledge with word embedding technology. Throughout this study the basic elements of the model for text document clustering are presented. The designed system consists of document preprocessing module (tokenization, normalization, stop-word removal and stemming) that is essential for better extraction. Structuring encyclopedic knowledge module, consists of representations of categorical concepts and the tree like linked representation of these concepts. Feature extraction extracts representative features relevant to the original text sets, so as to reduce the dimensionality of feature vector spaces. This module includes relating text documents with encyclopedic knowledge and finding the representative features of the text (categorical concepts, related categorical concepts). Learning word embedding (word2vec) vector representation of text document captures the distributional representations of words. The contextual terms of the related concept features are extracted using neural word embedding technique, word2vec.

One element of this study was structuring Encyclopedic knowledge in the form that can be used for text feature extraction. We used tree like database structure ltree by taking Wikipedia (online encyclopedia). Wikipedia contains categorical topic labels that have tree like data structure to each document and these are used as knowledge for different text analytics tasks. In this study, Amharic Wikipedia database dump is structured and used for testing. In Amharic language there are a number of multi word terms like hyphenated and abbreviated words. Hence some of these terms were identified for testing to keep the semantics of the documents.

The context of a term in natural language text is given by the interconnection of the different words employed in a sentence for which the semantics of each word is known. Word2vec is neural network based word embedding model that can establish similarities between terms. In this study, we enriched text document features using the contexts of related categorical concepts and most probable contextual word of concept features based on trained word2vec model. Text features are extracted for each text document using encyclopedic knowledge with neural word embedding technology. The importance of the

feature is evaluated using text weighting process. Uncorrelated features are eliminated to make more weighted concepts more descriptive by providing weight threshold. Among several k-means algorithms available, in this study, we used spherical k-means algorithm for clustering text documents. The spherical k-means algorithm, i.e., the k-means algorithm with cosine similarity, is a popular method for clustering high-dimensional text data. By collecting K classes of text documents, we evaluated the clustering results using the evaluation matrices precision, recall and accuracy.

The experimental results demonstrate that our model can achieve new improved performances on text clustering task. As shown on testing, these techniques and processes used in unsupervised text document clustering using encyclopedic knowledge with neural word embedding, the result of correctly clustered documents is 94.95% of the total test documents, which shows the average accuracy of the clustering is good. Furthermore, having enriched encyclopedic knowledge, improved word embedding technology and capability of linguistic preprocessing tools decides the final result of clustering text documents. Thus as the encyclopedic knowledge base gets richer, the performance of the system will be improved significantly.

6.2 Contribution of the Study

The main contributions of the study are listed below:

- A generic model is designed for unsupervised text document clustering that takes advantage of encyclopedic knowledge and neural word embedding technology.
- This study showed the difference between clustering text documents using encyclopedic knowledge with word embedding and without word embedding.
- This study contributes the algorithm for feature enriched representation of text documents by using encyclopedic knowledge with word embedding.
- This study showed the possibility of the conceptual relation between text documents to be clustered based on their distribution during clustering.
- This study showed the association between the cluster size and the accuracy of the designed text clustering model.
- In addition, the study contributes to the growth of semantic feature extraction to text data analysis using advantages of encyclopedic knowledge and word embedding.

6.3 Future Works

The designed system, i.e., unsupervised text document clustering using encyclopedic knowledge with word embedding explores and attempts improved text document organization by considering semantics of documents. However, it is also learnt that further research and developmental effort is needed so as to enable text data analysis and organization more accurate. Furthermore, there are some components that should be enriched and integrated for better functioning of the system. Some of the future research issues and features that needed to provide a better result include:

- Acquiring good semantic representation of text document needs enriched knowledge base, because texts written in different natural languages have ambiguous terms or sentences that affects in acquiring semantics. It would be better to use by integrating different knowledge base designed like expert, geographic and other knowledge bases with encyclopedic knowledge in semantic feature extraction.
- Using encyclopedic knowledge with word embedding technologies for different data mining tasks is believed to result in a significant improvement of the task. In this study we tried only text document clustering. However, it is also believed to result a significant improvement in text summarization, topic modeling, text classification, time serious event analysis and other text mining tasks.
- It is interesting to validate the effectiveness of using tagged concepts and lexical database that offers information related to various semantic relationships among words would be essential. It has additional potential that would be used with extracted features. Having tagged text feature with its database information is believed to result in a significant improvement of the clustering process.

References

[1] Ahammad Fahad and Wael M.S. Yafooz, "Review on Semantic Document Clustering", *International Journal of Contemporary Computer Research (IJCCR),* March 2017.

[2] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Soren Auer, and Christian Bizer, DBpedia – "A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia, *Journal of Semantic Web*, 2015.

[3] Wikipedia free encyclopedia, retrieved from *https://en.wikipedia.org,* Last accessed on October 25, 2017.

[4] Yohannes Afework, "Automatic Amharic Document Categorization: The Case of Ethiopian News Agency", Unpublished Master's Thesis, Department of Computer Science, Addis Ababa University, 2013.

[5] Mulualem Wordofa, "Semantic Indexing and Document Clustering for Amharic Information Retrieval", Unpublished Master's Thesis, School of Information Science, Addis Ababa University, 2013.

[6] Internet Usage World Statistics, "Internet and Population Statistics 2017", retrieved from http://www.internetworldstats.com/, Last accessed on November 22, 2017.

[7] Manoj Kumar, D. K. Yadav, and Vijay Kumar Gupta, "Frequent Term Based Text Document Clustering: A New Approach", IEEE *International Conference on Soft Computing Techniques and Implementations (ICSCTI),* 13 June 2016.

[8] JunkaiYi, Yacong Zhang, Xianghui Zhao, and Jing Wan, "A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network", *Hindawi Publishing Corporation,* 15 March 2017.

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

[10] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, E. K. Park and Xiaohua Zhou, "Exploiting Wikipedia as External Knowledge for Document Clustering", *Semantic Scholar,* June 2009.

[11] Nagma Y. Saiyad, Harshadkumar B. Prajapati, and Vipul K. Dabhi, "A Survey of Document Clustering using Semantic Approach", *International Conference on*

Electrical, Electronics, and Optimization Techniques (ICEEOT) IEEE, March 2016.

[12] Abdullah G., Aisha Siddiqa, Shahaboddin Shamshirb, and Fariza Hanum, *"*A survey on Indexing Techniques for Big Data*", Springer, Verlag*, London, 2015.

[13] PennState Eberly College of Science Online Courses, retrieved from https://onlinecourses.science.psu.edu/stat505, Last accessed on January 1, 2018.

[14] Mesfin Abate and Yaregal Assabie, "Development of Amharic Morphological Analyzer Using Memory-Based Learning", *Springer International Publishing, Switzerland*, 2014.

[15] Tomas Mikolov, Ilya Sutskever, and Kai Chen, "Distributed Representations of Words and Phrases and their Compositionality", *Cornell University Library*, New York, 2013.

[16] Towards Data Science, Word Embeddings, retrieved from *https://towardsdatascience.com/,* Last accessed on *January 21,2018.*

[17] Hu X and Liu H, "Text Analytics in Social Media", in Aggarwal C., Zhai C, Mining Text Data, Springer, Boston, MA, 2012.

[18] Anna Huang, David Milne, Eibe Frank, and Ian H.Witten, "Clustering Documents with Active Learning using Wikipedia", *Eighth IEEE International Conference on Data Mining*, February 10, 2009.

[19] Alaa Alahmadi, Arash Joorabchi, and Abdulhussain E. Mahdi, "A New Text Representation Scheme Combining Bag-of-Words and Bag-of-Concepts Approaches for Automatic Text Classification", *IEEE GCC Conference and Exhibition*, November 20, 2013.

[20] Ndargachew Mekonnen, "Automatic Thesaurus Construction for Amharic Text Retrieval", Unpublished Master's Thesis, School of Information Science, Addis Ababa University, 2009.

[21] Wael H. Gomaa and Aly A. Fahmy, "A Survey of Text Similarity Approaches", *International Journal of Computer Application*s, April 2013.

[22] Anna Huang, "Similarity Measures for Text Document Clustering", *Proceedings of the 6th Computer Science Research Student Conference*, New Zealand, January 2008.

[23] N. Sandhya, Y.Sri Lalitha, A.Govardhan, and K.Anuradha, "Analysis of Similarity Measures for Text Clustering". *Semantic Scholar*, 2013.

[24] Michael Steinbach, George Karypis, and Vipin Kumar, “A Comparison of Document Clustering Techniques”, *Proceedings of the International KDD Workshop on Text Mining*, June 2000.

[25] Fidan Kaya Gülağız and Suhap Şahin, “Comparison of Hierarchical and Non Hierarchical Clustering Algorithms”, *International Journal of Computer Engineering and Information Technology*, January 2017.

[26] Data Science, K-means Clustering ,retrieved from *https://www.datascience.com* , Last accessed on, January 2018.

[27] Shi Zhong, “Efficient Online Spherical K-means Clustering”, *IEEE International Joint Conference on Neural Networks*, August 2005.

[28] Shenghong Yang and Yongheng Wang, “Density-Based Clustering of Massive Short Messages using Domain Ontology”, in *Asia-Pacific Conference on Information Processing*, May 2009.

[29] Peiyu LIU, Yingying Liu, Xiuyan Hou, Qingqing Li, and Zhenfang Zhu, “Text Clustering Algorithm Based on Find of Density Peaks”, in *7th International Conference on Information Technology in Medicine and Education*, 2015.

[30] Stuti Karol and Veenu Mangat, “Evaluation of Text Document Clustering Approach Based on Particle Swarm Optimization”, *Central European Journal of Computer Science*, September 2012.

[31] Lailil Muflikhah and Baharum Baharudin, “Document Clustering Using Concept Space and Cosine Similarity Measurement”, *International Conference on Computer Technology and Development*, 2009.

[32] Mingyu Yao, Dechang Pi, and Xiangxiang Cong, “Chinese Text Clustering Algorithm Based on K-means”, in *International Conference on Medical Physics and Biomedical Engineering*, 2012.

[33] Andrey Kutuzov, Mikhail Kopotev, Tatyana Sviridenko, and Lyubov Ivanova, “Clustering Comparable Corpora of Russian and Ukrainian Academic Texts: Word Embeddings and Semantic Fingerprints”, *Semantic Scholar*, Apr 2016.

[34] Venkata Srikanth Reddy, Patrick Kinnicutt, and Roger Lee, “Text Document Clustering: The Application of Cluster Analysis to Textual Document”, *International Conference on Computational Science and Computational Intelligence*, 2016.

[35] Manoj Kumar, D. K. Yadav, and Vijay Kumar Gupta, “Frequent Term Based Text Document Clustering: A New Approach”, IEEE *International Conference on Soft Computing Techniques and Implementations (ICSCTI)*, 13 June 2016.

[36] Cheng-Lin Yang, Nuttakorn Benjamasutin, and Yun-Heh Chen-Burger, “Mining Hidden Concepts: Using Short Text Clustering and Wikipedia Knowledge”, *IEEE 28th International Conference on Advanced Information Networking and Applications Workshops*, 26 June 2014.

[37] Yiming Li, Baogang Wei, Liang Yao, Hui Chen, and Zherong Li, “Knowledge-based Document Embedding for Cross-Domain Text Classification”, *Institute of Electrical and Electronics Engineers (IEEE)*, 2017.

[38] Hanane Froud and Abdelmonaime Lachkar, “Agglomerative Hierarchical Clustering Techniques for Arabic Documents”, *Springer*, Switzerland, September 2013.

[39] Fawaz S. Al-Anzi and Dia AbuZeina, “Big Data Categorization for Arabic Text Using Latent Semantic Indexing and Clustering”, *International Conference on Engineering Technologies and Big Data Analytics*, Bangkok, January 2016.

[40] Abegaz, Yalemsew, “Document Clustering in Amharic”, in *Third Workshop on African Language Technology*, December 2011.

[41] Samuel Eyassu, Lars Asker, Atelach Alemu, Bjorn Gamback, and Lemma Nigussie, “Classifying Amharic Web News”. *Springer Science Business Media*, February 2009.

[42] Seffi Gebeyehu1 and Vuda Sreenivasa Rao,” A Two Step Data Mining Approach for Amharic Text Classification”, *American Journal of Engineering Research (AJER)*, 2014.

[43] Worku Kelemework, “Automatic Amharic Text News Classification: Neural Networks Approach”, *Ethiopian Journal of Science and Technology*, 2013.

[44] Abraham Hailu and Yaregal Assabie, “Itemsets-Based Amharic Document Categorization Using an Extended Apriori Algorithm”, *Springer International Publishing*, Switzerland 2016.

[45] Alemu Kumilachew Tegegnie, Adane Nega Tarekegn, and Tamir Anteneh Alemu,"A Comparative Study of Flat and Hierarchical Classification for Amharic News Text Using SVM", *International Journal of Information Engineering and Electronic Business(IJIEEB)*, Vol. 9, No. 3, 2017.

[46] Alemayehu, Nega, and Willett Peter, "Stemming of Amharic Words for Information Retrieval", *Literary and Linguistic Computing*, January 2002.

[47] Pu Han, Dong-bo Wang, and Qing-guo Zhao, "Chinese Document Clustering Based on Weka", *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*, Guilin, July 2011.

Annex A: List of Stop-words

ሁሉ	በኩል	እባክዎ	ውጪ	አርስ	አንስተዋል	ኋላ
ሁሉም	በጣም	አንድ	ያለ	ተችሏል	አብራርቷል	ተችተዋቸዋል
ኋላ	ብቻ	አንጻር	ያሉ	አስይዟል	ተዘዋውረዋል	ተቀይረዋል
ሁኔታ	በተለይ	እስኪደርስ	ይገባል	ተናግሯል	ሊሆን	ተጠቅሟል
ሆነ	በተመለከተ	እንኳ	የኋላ	ያሳስበዋል	አልገቡም	አልነበረም
ሆኑ	በተመሳሳይ	እስከ	የሰሞኑ	ተረክበዋል	ነግረዋቸዋል	ተሞክሯል
ሆኖም	የተለያየ	እዚሁ	የታች	ይወሳል	ቀድሞውም	ይህንንም
ሁል	የተለያዩ	እና	የውስጥ	ይባላል	አመለከት	አላስብም
ሁሉንም	ተባለ	እንደ	የጋራ	አይታወቅም	አልተፈጠረም	እርሱም
ላይ	ተገለጸ	እንደገና	ይታወሳል	ተለቋል	ምክንያት	እዚያ
ሌላ	ተገልጿል	እንዲሁም	ይህ	አስቀምጧል	ታልፏል	ይገለጻል
ሌሎች	ተጨማሪ	እንጂ	ደግሞ	ሰጥቷቸዋል	ምንል	ቻሉት
ልዩ	ተከናውኗል	እዚህ	ድረስ	አታደርግም	አሞግሷል	ይስተዋላል
መሆኑ	ታች	እዚያ	ጋራ	አያስከትልም	አንድም	አናስታውስም
ማለት	ትናንት	እያንዳንዱ	ግን	ይናገራል	አሰምቷል	ያቀርባል
ማለቱ	ነበረች	ኋላ	ገልጿል	አፍርሰናል	አውግዟል	ተፈጥሯል
መካከል	ነበሩ	ከላይ	ገልጸዋል	አልተሞከረም	ይታያል	ሰፍሯል
የሚገኙ	ነበረ	ከመካከል	ግዜ	ሌሎችም	ይፈቀድላቸዋል	ትናንት
የሚገኝ	ነው	ከሰሞኑ	ደግሞ	ሰጥተውበታል	ይኼንንም	በርካታ
ማድረግ	ነይ	ከታች	ዛሬ	እንጂ	ያወሳል	እባክህ
ማን	ነገር	ከውስጥ	ጋር	አረጋግጠዋል	ደርሳል	አስከትሏል
ማንም	ነገሮች	ከጋራ	ተናግረዋል	ሁለቱም	ይገኙባቸዋል	አጠናቋል
ሰሞኑን	ናት	ከፊት	የገለጹት	ቀድም	አስተለልፈዋል	የጋራ
ሲሆን	ናቸው	ወዘተ	ይገልጻል	ሆኖኛል	ሆና	ችሎት
ሲል	አሁን	ወይም	ሲሉ	ይኖርብናል	ይታወቃል	በእነዚህ
ሲሉ	አለ	ወደ	ብለዋል	አይኖርም	ተወስኗል	ዋና
ስለ	አስታወቀ	ዋና	ስለሆነ	እንኳ	አልተጠናቀቀም	ግን
ቢሆን	አስታውቀዋል	ወደፊት	አቶ	አምነዋል	አይቀርብም	ወደ
ብለዋል	አስታውሰዋል	ሲል	ሆኖም	ቀርባቸዋል	አውጥተዋል	ተፈጽሟል
ብቻ	እስካሁን	በቀር	አመልክተዋል	አያውቅም	ከፊት	ቋም
ብዛት	አሳሰበ	በፊት	ይናገራሉ	አስቀምጠዋቸዋል	ተደግፏል	ያምናል
ብዙ	አሳስበዋል	ውስጥ	አበራርተው	ተወስተዋል	ስላሉ	ቆይታል
ቦታ	አስፈላጊ	እባክሽ	አስረድተዋል	ያትታል	ናቸው	ይመሠርታል
በርካታ	አስገነዘቡ		እስከ		ከጋራ	በጣም

በሰሞኑ	አስገንዝበዋል	እያንዳንዳቸው	ተጠቅሷል	ብለዋል	ክትትል	በውኑ
በታች	አብራርተዋል	አካሂደዋል	ያባብሱታል	አልነበሩም	መሆኑም	አይዘነጋም
በኋላ	ትናገራል	ተደርገዋል	ይጭራል	በታች	ሳይሆን	አስይዘዋል
እባክህ	ሰሞን	ይጠቅሳል	ተንትኗል	ተዳርገዋል	ማንኛውም	ተሠርተዋል
አይመስለኝም	ሆናል	ወዘተ	ይገኙበታል	ስለዚህ	ተሰጥቶበታል	ብአድ
አስታውሷል	ቆይተዋል	አይደለምን	ተሰምቷል	ይኖሩታል	ተደርጓል	ወደዚህ
አይገምቱም	አምኗል	ነበራች	አቶ	ተዘርዝሯል	ነህ	አቅርበዋቸዋል
ይኖረዋል	በዚያም	አመልክተዋል	ተነግሯል	ነን	ተቃጥሏል	ተረጋግጧል
እላችኋለሁ	ያመላክታል	አሳውቋል	ተበትኗል	አሳስቧል	ሌላ	ሆነ
አጠናቀዋል	ይቀጥላል	ተከስተዋል	እነዚህንም	ይመከራል	አልቀረቡም	ብላችሁ
አልቀረበም	ነገሮች	አልተገለጸም	ቢል	ነኝ	ይሁን	ሆኑ
ሆይ	ናት	አስቀምጦታል	አስጠንቅቋል	ዳሰዋል	ተሰጥተዋል	እያንዳንዷ
አልታየም	አከለዋል	ይታወሳል	ተጠቆም	አድንቀዋል	አለው	ሰሞኑ
አስገብተዋል	አይችሉም	እናንተ	አድርጓቸዋል	በፊት	እስከ	አሉም
አትቷል	ተሰጥቷቸዋል	ያከናውናል	በሆነ	ተቆጥረዋል	አግኝተዋል	ተሰንዝረዋል
ይከፍታል	ይቋቋማል	ውስጥ	አይደሉም	አብቅቷል	አይቀበሉትም	ጠፍቷል
የለም	ሆነም	ኖርን	አስታውቋል	ተቋርጧል	ወጥቷል	በዚያው
ይሠራል	እባክዎ	በሆላ	ይቻላል	አላገኙም	ተቆጥበዋል	አስተባብለዋል
ይገርመኛል	አላደረገም	አህ	እነዚያም	ወይዘሪት	ተዘግቧል	አውስተዋል
ሆነህ	ተቆጥሯል	ተጠቅመዋል	አንችልም	ይሰጣቸዋል	ይገኝበታል	ጋይተዋል
ከኋላ	አላቆሙም	አሉ	ተይቀዋል	ሆኗል	አስተየት	ማናቸውንም
ተነስቷል	ይችላል	ቢሆን	ገብታል	ከላይ	ተለያዩ	ከዚያ
ማንም	አስተላልፈዋል	አስተላልፏል	ተወያይተዋል	ይገምታል	መግለጭ	አስተባብሯል
ይኖርበታል	ተሰጥታል	አካቷል	ባሉ	አቅርቧል	መርጧቸዋል	አለብአወጥ
ያገለግላል	አሳይተዋል	ቀጥለዋል	ሰሞኑን	አይቻልም	ዘርዝረዋል	እኔ
ብለውታል	ተመልሰዋል	አሳስበዋል	በውስጥ	አራምደዋል	ተመላክቷል	አልችልም
ይጠፉዋል	አድርጋዋል	ይሰጣል	ተነግሯቸዋል	አግኝታል	ያሳስባል	አልችልም
ተጠቁሟል	ሆነው	አንፈልግም	ቢኖር	አደርጓል	አነጋግረዋል	ቀበላቸዋል
አድርጎታል	ከመካከል	አንፀባርቀዋል	ያመለክታል	ሆኗልም	አልፏል	አልጀመሩም
አስታውሰዋል	ሆኖም	ነበረች	ነበር	አልተካተተም	እያንዳንዷ	አልቻሉም
ነች	ይጠይቃል	ይሰማል	መካከል	ተገብቷል	አለኝ	ተመሠረት
አሁንም	ቆይቷል	በሌላ		ትሆን	በዚህ	አያስፈልግም
ታውቋል	ያሳያል	እንደገለጹት		እባክሽ	አሉት	አድርገናል
አብራርተዋል	አለች	እንደአስረዱት		አይችልም		አቅርበዋል
ለማንም		ሌላ		ሆንሁ		ቢገለጹም
ያለ				እባኩዎ		

Annex B: List of Normalized Alphabets

[\\ሃ\\ኅ\\ሀ\\ኃ\\ሐ\\ሓ\\ኻ] → ሀ

[\\ሁ\\ሑ\\ኁ\\ዅ] → ሁ

[\\ሂ\\ኂ\\ሒ\\ኺ] → ሂ

[\\ሄ\\ኄ\\ሔ\\ኼ] → ሄ

[\\ህ\\ሕ\\ኅ] → ህ

[\\ሆ\\ኆ\\ሖ\\ኾ] → ሆ

[\\ስ\\ሠ] → ሰ

[\\ሱ\\ሡ] → ሱ

[\\ሲ\\ሢ] → ሲ

[\\ሳ\\ሣ] → ሳ

[\\ሴ\\ሤ] → ሴ

[\\ስ\\ሥ] → ስ

[\\ሶ\\ሦ] → ሶ

[\\አ\\ዓ\\ኣ\\ዐ] → አ

[\\ኡ\\ዑ] → ኡ

[\\ኢ\\ዒ] → ኢ

[\\ዔ\\ኤ] → ኤ

[\\እ\\ዕ] → እ

[\\ኦ\\ዖ] → ኦ

[\\ጸ\\ፀ] → ፀ

[\\ጹ\\ፁ] → ፁ

[\\ጺ\\ፂ] → ፂ

[\\ጻ\\ፃ] → ፃ

[\\ጼ\\ፄ] → ፄ

[\\ጽ\\ፅ] → ፅ

[\\ጾ\\ፆ] → ፆ

Annex C: List of Amharic Abbreviations from collected corpus

ህ/ተ/ም/ቤት	ህዝብ እንደራሴዎች ምክር ቤት
ህ/ተ/ም/ቤት	ህዝብ ተወካዮች ምክር ቤት
ህወሃት	ህዝባዊ ወያኔ ሃርነት ትግራይ
ሃ/አለቃ	ሀምሳ አለቃ
ሊ/መንበር	ሊቀ መንበር
ሌ/ኮለኔል	ሌተናል ኮለኔል
መኢብን	የመላው ኢትዮጵያ ብሔራዊ ንቅናቄ
መኢአድ	የመላው ኢትዮጵያ አንድነት ድርጅት
መኢዴፓ	የመላው ኢትዮጵያ ዴሞክራሲያዊ ፓርቲ
መኦህዴፓ	የመላው ኦሮሞ ህዝብ ዴሞክራሲያዊ ፓርቲ
መ/ር	መምህር
መ/ቤት	መስሪያ ቤት
ሚ/ር	ሚኒስትር
ሜ/ጀነራል	ሜጀር ጀነራል
ም/ቤት	ምክር ቤት
ተመድ	የተባበሩት መንግስታት ድርጅት
ተ/ሃይማኖት	ተክለ ሃይማኖት
ት/ቤት	ትምህርት ቤት
ጠ/ሚኒስትር	ጠቅላይ ሚኒስትር
ካፍ	አፍሪካ እግር ኳስ ኮንፌዴሬሽን
ክ/ሀገር	ክፍለ ሀገር
ክ/ከተማ	ክፍለ ከተማ
ወ/ር	ወታደር

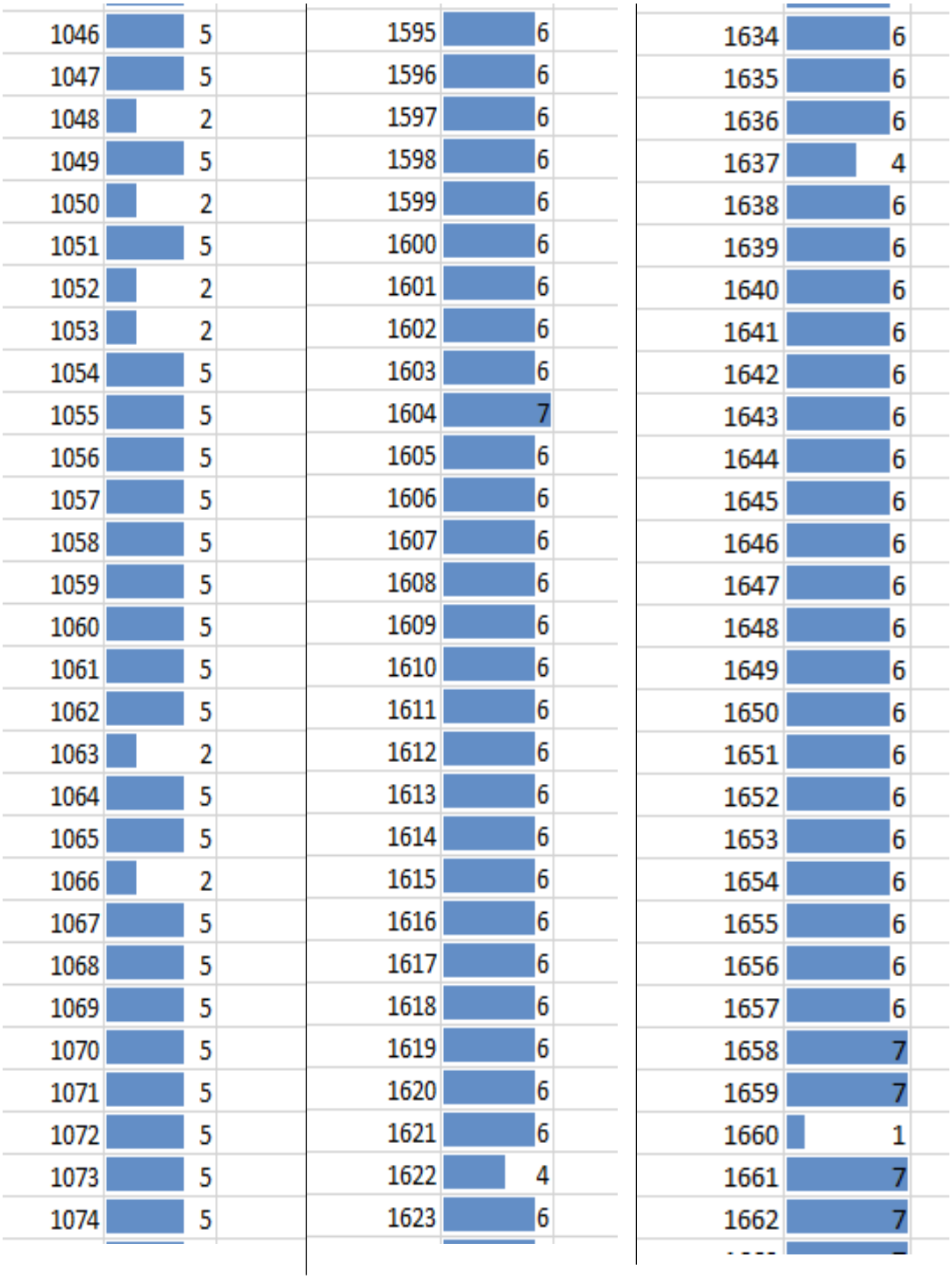
ወ/ሮ	ወይዘሮ
ወ/ሪት	ወይዘሪት
ፍ/ቤት	ፍርድ ቤት
ጽ/ቤት	ጽህፈት ቤት
ዶ/ር	ዶክተር
ቤ/ክርስትያን	ቤተ ክርስትያን
ኮ/ል	ኮለኔል
ብ/ጀነራል	ብርጋዴር ጀነራል
አ/አ	አዲስ አበባ
ኦፌኮ	ኦሮሞ ፌደራሊስት ኮንግረስ
ፌ/ ም/ ቤት	ፌዴሬሽን ምክር ቤት
ኢህአዴግ	ኢትዮጵያ ህዝቦች አብዮታዊ ዴሞክራሲያዊ ግንባር
ብአዴን	ብሔረ አማራ ዴሞክራሲያዊ ንቅናቄ
ኢዴፓ	የኢትዮጵያዊያ ዴሞክራሲያዊ ፓርቲ
ኦህዴድ	ኦሮሞ ህዝቦች ዴሞክራሲያዊ ድርጅት
ኢሶዴፓ	የኢትዮጵያ ሶሻል ዴሞክራሲ ፓርቲ
አረና	አረና ትግራይ ለዴሞክራሲና ሉዓላዊነት
ኢህዴን	ኢትዮጵያ ህዝብ ዴሞክራሲያዊ ንቅናቄ
ደኢህዴን	የደቡብ ኢትዮጵያ ህዝቦች ዴሞክራሲያዊ ንቅናቄ
ቅንጅት	ቅንጅት ለአንድነትና ለዴሞክራሲ ፓርቲ
ቻን	አፍሪካ አገሮች ዋንጫ
ኢዴፓ	የኢትዮጵያውያን ዴሞክራሲያዊ ፓርቲ
ኢዴህ	የኢትዮጵያ ዴሞክራቲክ ኀብረት
ፕ/ት	ፕሬዝዳንት
ዓ/ም	አመተ ምህረት

Annex D: Sample Output of Clustering

No. of Clusters =	7
Document ID	Cluster ID (1 - 7)
1	1
2	2
3	3
4	1
5	4
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	5
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1

47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1
61	1
62	2
63	1
64	1
65	1
66	1
67	1
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1

134	4
135	1
136	1
137	1
138	1
139	1
140	1
141	1
142	1
143	1
144	1
145	1
146	1
147	1
148	1
149	1
150	1
151	1
152	1
153	1
154	1
155	1
156	1
157	1
158	1
159	1
160	1
161	1
162	1


1046 5
1047 5
1048 2
1049 5
1050 2
1051 5
1052 2
1053 2
1054 5
1055 5
1056 5
1057 5
1058 5
1059 5
1060 5
1061 5
1062 5
1063 2
1064 5
1065 5
1066 2
1067 5
1068 5
1069 5
1070 5
1071 5
1072 5
1073 5
1074 5
1595 6
1596 6
1597 6
1598 6
1599 6
1600 6
1601 6
1602 6
1603 6
1604 7
1605 6
1606 6
1607 6
1608 6
1609 6
1610 6
1611 6
1612 6
1613 6
1614 6
1615 6
1616 6
1617 6
1618 6
1619 6
1620 6
1621 6
1622 4
1623 6
1634 6
1635 6
1636 6
1637 4
1638 6
1639 6
1640 6
1641 6
1642 6
1643 6
1644 6
1645 6
1646 6
1647 6
1648 6
1649 6
1650 6
1651 6
1652 6
1653 6
1654 6
1655 6
1656 6
1657 6
1658 7
1659 7
1660 1
1661 7
1662 7

Declaration

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

Declared by:

Name: __Dessalew Yohannes Bogale___

Signature: _____

Date: _____

Confirmed by advisor:

Name: ___Yaregal Assabie (PhD)_____

Signature: _____

Date: _____