ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Amharic Information Retrieval Using Semantic Vocabulary

Berihun Getnet Akalu

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

Addis Ababa, Ethiopia

October 2019

Addis Ababa University

College of Natural Sciences

Berihun Getnet Akalu

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by *Berihun Getnet Akalu*, titled: *Amharic Information Retrieval Using Semantic Vocabulary* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

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

Abstract

The increase in large scale data available from different sources and the user's need access to information retrieval becomes more focusing issue these days. Information retrieval implies seeking relevant documents for the user's queries. But the way of providing the queries and the system responds relevant results for the user should be improved for better satisfaction. This can be enhanced by expanding the original queries from semantic lexical resources that are constructed either manually or automatically from a text corpus. But, manual construction is tedious and time-consuming when the data set is huge. The way semantic resources are built also affects retrieval performance. Based on formal semantics the meaning is built using symbolic tradition and centered around the inferential properties of languages. It is also possible to automatically construct semantic resources based on the distribution of the word from unstructured data which applies the notion about unsupervised learning that automatically builds semantics from high dimensional vector space. This produces contextual similarity via word's angular orientation. There have been attempts done to enhance information retrieval by expanding queries from semantic resources for non-Ethiopian languages.

In this study, we propose Amharic information retrieval using semantic vocabulary. It is figured out by considering components including text preprocessing, word-space modeling, semantic word sense clustering, document indexing, and searching. After the Amharic documents are preprocessed the words are vectorized on a multidimensional space using Word2vec based on the notion words surrounding another word can be contextually similar. Based on the word's angular orientation, the semantic vocabulary is constructed using cosine distance. After Amharic documents are preprocessed it is indexed for later retrieval. Then the user provides the queries and the system expands the original query from the semantic vocabulary. The queries are reformulated and words are searched from indexed data that returns more relevant documents for the user.

A prototype of the system is developed and we have tested the performance of the system using Amharic documents collected from Ethiopian public media. The semantic vocabulary based on the word analog prediction using the cosine metric is promising. It is also compared against the semantic thesaurus constructed with the latent semantic analysis and it increases by 17.2% accuracy. Information retrieval using semantic vocabulary based on ranked retrieval increases by 24.3% recall, and using unranked set of retrieval, 10.89% recall improvement was obtained.

Keywords: Word2Vec, distributional semantics, semantic vocabulary, information retrieval.

Dedication

This thesis is dedicated to the memory of my lovely uncle Walelgn Demssie.

Acknowledgment

First and foremost, I would like to thank the Almighty God for giving me health, strength, knowledge, ability, and opportunity to undertake this research study, to persevere and complete. Without his blessings, this achievement would not have been possible.

Next, I would like to express my sincere gratitude to my advisor Yaregal Assabie (PhD) for his continuous support of this research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank Tilahun Yeshambel (PhD) candidate in information technology at Addis Ababa University for his expert judgment of relevant documents and his brotherly supporting, advising and initiative. Let God bless you all my best friends and classmates for your help, idea sharing and open communication which enabled me to complete the thesis.

Last but not least, I would like to thank my families for their love and support for each movement of time by any means, anytime, anywhere.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

API	-	Application Program Interface
BGN	-	Board on Geographic Names (USA)
BM25F	-	Best Matching Version 25F
CBOW	-	Continuous Bag of Words
DBOW	-	Distributed Bag of Words
DMPV	-	Distributed Memory Paragraph Vector
Doc2Vec	-	Document Vector
DSMs	-	Distributional Semantics Models
ESA	-	Explicit Semantic Analysis
HTML	-	Hyper-Text Markup Language
KNN	-	K-Nearest Neighbor
LSA	-	Latent Semantic Analysis
NLP	-	Natural Language Processing
NNM	-	Non-Negative Matrix Factorization
NumPy	-	Numerical Python
PCA	-	Principal Component Analysis
PCGN	-	The Permanent Committee on Geographical Names
QE	-	Query Expansion
RI	-	Random Indexing
SG	-	Skip Gram
SOV	-	Subject Object Verb
SVD	-	Singular value decomposition

TF-IDF	-	Term Frequency-Inverse Document Frequency
t-SNE	-	Distributed Stochastic Neighbor Embedding
URL	-	Uniform Resource Locator
VSR	-	Vector Space Representation
Word2Vec	-	Word to Vector
WSM	-	Word-Space Model

Chapter One: Introduction

1.1 Background

Natural language processing is the ability of a computer system to understand human languages as we speak and communicate with each other. It is an interdisciplinary field of study dealing with computational techniques for analyzing and representing naturally occurring texts. It is an active research area that explores how computers can be used to understand and manipulate natural language texts [1]. Natural language processing is done with different levels of understanding like phonetical, morphological, pragmatic, discourse, syntactic, and semantics in which the information retrieval system is built upon it. Now a day's information retrieval based on semantics becomes more popular due to the language nature and understanding of meaning is subjective. Information retrieval is the process of seeking relevant documents from a certain source that can be a database, indexed file or web page. The performance of an IR system may be weak and to satisfy the user's need and guarantee their searching must be enhanced using some ready-made lexical resources that are built automatically from a text corpus by expanding the user's query. Query expansion is the process of reformulating the original query into a set of additional multiple similar terms to retrieve more relevant documents from data sources.

The increasing of large corpora and high-speed processing power of computers led to a growing interest in automation and data-driven linguistic analysis. Using natural language texts lexical resources like semantic vocabulary, thesaurus, dictionary, and WordNet can be constructed either manually or automatically. For example, semantic vocabulary can be constructed automatically from a corpus based on the assumption words distributed across a multidimensional vector space could have the same meaning. The similarity measure can be compared via mathematical functions [2].

Semantics is the study of meaning which is a wide subject within the general study of languages. It investigates how to extract meaning from a given text and the way that the meaning of words can be combined to imply the meaning of large corpus [3]. There are different approaches that can identify the linguistic meaning for a given text. These are formal semantics and distributional semantics.

Formal semantics is different from distributional semantics in a way that, it is based on a symbolic tradition and centered around the inferential properties of language whereas the distributional semantics is statistical and data-driven, and focuses on aspects of meaning related

to descriptive content [4]. In this concept, the meaning of words can be represented from word contexts in which it appears and can be formalized a vector in multi-dimensional distributional space.

Distributional semantics is one approach of semantics in which the meaning is extracted from word contexts and its distributions across a vector space. It is a theory of meaning which is computationally implementable and very good at modeling what humans do when they make similarity judgment. It models computational representation of word meaning from the patterns of co-occurrence of words in each text which can provide reliable estimates of meaning relatedness for many semantic tasks requiring them. It extracts meaning information exclusively from the text which states words that occur in similar contexts are semantically similar [5].

The hypothesis behind the distributional semantics states that at least certain aspects of the meaning of lexical expressions depend on their distributional properties in the linguistic contexts. For instance, the degree of semantic similarity between two linguistic expressions α and β is a function of the similarity of the linguistic contexts in which α and β can appear. The distributional similarity across a vector space may be weak or strong. If the similarity distance is weak then the words are far apart and it does not further wanted for linguistic analysis. Strong view as a cognitive hypothesis about the form and origin of semantic representations; if word distributions in context play a specific causal role in forming meaning representations. This assumes that there exists a strong correlation between the observable distributional characteristics of a word and its meaning [6].

Distributional semantic models (DSMs) are computational methods that turn the distributional hypothesis into an experimental framework for semantic analysis which is extracted from corpora and count co-occurrences of lexical items with linguistic contexts represent lexical items geometrically with distributional vectors built out of their co-occurrence counts and measures semantic similarity with distributional vector similarity [7]. There is also another meaning similarity for words like WordNet. However, WordNet contains a smaller number of words with its relation and similarity where distributional semantics contains many words with similar meaning and multiple relations. WordNet provides a typed semantic space whereas DS provides automatically created vector space. The performance of distributional semantics is better than WordNet because of large corpora with multiple dimensions. Word2Vec returns more hypernyms, synonymy, and hyponyms which shows the probability that neighbors

separated by a given cosine distance are semantically related and many relations can be constructed [8].

Words that are said to be similar must refer to the same contextual meaning and the similarity distance across the vector space must be nearly related. An M by N dimension is extracted from Amharic document by which M represents words and N represents contexts. Words are clustered based on their distributional similarity together in multi-dimensional space.

Once a lexical resource is built it is used for the information retrieval system by expanding the query words automatically to other similar words and searches the reformulated terms from indexed data sources. The users are not aware of the expanded query terms but the enhanced information retrieval system reformulates the query from the semantic vocabulary, searches from the indexed sources, and finally returns more relevant documents.

1.2 Motivation

The meaning of words can vary according to the contexts in which it appears and this may be understandable by humans but may be difficult and requires an efficient model for a computer system to understand. Manual construction of vocabularies from large corpora takes time and it is bulky to process. Unless the meaning of words can be interpreted via its context it may become difficult to understand. Retrieving relevant documents using semantic vocabulary as a query expansion can be used for information retrieval. To the best of our knowledge, no research work is done for constructing semantically related vocabulary from Amharic corpus using distributional semantics and its use for information retrieval. This has motivated us to conduct research on Amharic information retrieval using semantic vocabulary.

1.3 Statement of the Problem

The need access to information retrieval using query expansion from a lexical resource like WordNet, dictionary, and thesaurus becomes a focus research area these days. These resources can enhance the retrieval relevancy, but the nature of meaning extraction affects the performance. Semantic vocabulary using the neural word embedding method is the new approach for huge data size and it can be used for information retrieval by expanding the original queries into reformulated semantic similar queries. There have been attempts done for constructing semantic resources for non-Ethiopian languages and it is discussed the application area where it can be applied. Distributional semantics is one approach of semantics which can

build meaning from a text. The notion “the company knows by the word it contains” works for distributional semantics. Words surrounding other words can have similar meanings.

Previous studies on distributional semantics for automatic and semi-automatic vocabulary construction and expansions for Japanese [9], Swedish [10], French [11], and English [12] languages showed promising results. Distributional semantics outperformed well for vocabulary construction for many languages so far. Some research works have also been carried out for Amharic semantics analysis [13] to show the semantic relations of words from a given corpus with the stored words in WordNet [14] by which contextually similar words may be ignored from relation sets because of the limited and predetermined number of words. Andargachew Makonnen [15] conducted research on automatic thesaurus construction for enhancing information retrieval using latent semantic analysis (LSA) with singular value decomposition (SVD) based on the words bag of count method. The meaning of words is extracted based on their frequency in the corpus that less frequent words are less informal for meaning information.

Understanding the limitation of previous studies on Amharic semantic analysis, we hypothesize that distributional semantics can perform better in representing the meaning of Amharic text, which in turn can be used for vocabulary construction. However, to the best of our knowledge, distributional semantics in general and its use in vocabulary construction has not been investigated for Amharic text. Because approaches depend on the characteristics of languages, previous studies conducted for other languages cannot be directly applied for Amharic. Thus, the aim of this study is to investigate the use of distributional semantics to construct semantically related vocabulary for Amharic information retrieval.

1.4 Objective

General Objective

The general objective of this research work is to automatically construct semantic vocabulary from Amharic text corpus using a distributional semantics approach for information retrieval.

Specific Objective

The specific objectives of this research work are the following:

- Conducting a literature review to understand Amharic semantics
- Collecting Amharic text corpus
- Designing Amharic information retrieval with query expansion
- Designing the Amharic word-space model
- Clustering Amharic words based on their semantic senses
- Constructing Amharic semantic vocabulary using neural word embedding
- Develop a prototype for the system
- Testing and evaluating the system

1.5 Methods

Literature Review

We have reviewed the papers done in the area of natural language processing especially using the distributional semantics. We assessed the techniques, the corpora and algorithms used to achieve the research objectives and how the meanings can be extracted from a certain document has been defined. It is also discussed the basic application area of semantic vocabularies and the gaps that different researchers achieved different results, by which the approaches used have a significant factor for the performances. As the contextual meaning extraction from text corpora becomes popular these days, we have clearly articulated the results achieved on different papers and all over state of the art about distributional semantics.

Data Collection

Many of the Amharic documents used for our work were collected from online free data portals of different domains like - Ethiopian reporter, Fana broadcasting corporate, Amharic bible, Ethiopian telecommunication, and z-habesha.com.

Prototype Development

To physically represent, illustrate and verify the aspects of conceptual design as part of the development process the prototype development is performed.

Implementation Tools

We have used java programing language for the Amharic stemmer. But our work other than stemming is totally implemented using python programing language for both semantic vector API, searching and indexing due to its performance when using huge data sizes. We have implemented our work using Spyder editor, Qt Designer, and whoosh document indexer.

Evaluation

The evaluation of our system is implemented in two ways. The first one is based on word analog and relatedness. The word analog distance is calculated using cosine values which is approaching 1 and this implies using cosine metric is better. We have also prepared fourteen multiple questions with each question containing a list of associated words and the answer is the associated score with a maximum of either five or a minimum of one. The semantic similar words in the question are trained in the latent semantic analysis [15] and word2vec**.** The result was calculated that is gained using LSA and word2vec.

The second evaluation was implemented by integrating the semantic vocabulary into an information retrieval system as query expansion. The information retrieval system is based on the ranked and unranked set of retrieval. Unranked retrieval has been expressed using the two basic retrieval relevance measures called recall and precision.

We have also evaluated the system using ranked retrieval using three common term scoring algorithms including frequency, TF-IDF, and BM25F. These scoring functions calculate the query score based on the Boolean searches and return the relevant document with descending order via its relevance. The BM25F with ORGroup Boolean search could return many relevant documents than the frequency-based retrieval. We have calculated the mean average recall and precision via its document ranks with semantic vocabulary as query expansion and without expansion. The result for the mean average recall is increased while using semantic vocabulary.

1.6 Scope and Limitation

This work deals with the automatic construction of semantically related vocabulary from Amharic corpus using distributional semantics for information retrieval and the contextual meaning of words can be defined at the word level of a large corpus. In general, the work is corpus dependent that considers only Amharic and does not work for other languages, it does not consider the parts of speech and phrasal words.

1.7 Application of Results

In addition, with query expansion and information retrieval semantic vocabulary is used:

- ✓ In lexicography, ontology and thesauri learning and population to automatically extract glossary terms from a large corpus based on the context that the word appears and to automatically define the meaning of each word.
- ✓ For word sense disambiguation and relation extraction. Hence each word is referenced by its context there is no chance of happening ambiguity.

1.8 Organization of the Thesis

The remaining section of the thesis is organized as Chapter Two presents the literature review in the domain of semantics and its use in information retrieval. It discusses the sciences and techniques used for contextual meaning extraction from a set of text documents. Chapter Three discusses related works in the area of semantic lexical resources for enhancing information retrieval, especially for semantic vocabulary construction. Chapter Four deals with the design of Amharic information retrieval. Chapter Five presents the implementation, prototype, and evaluation of our study. Finally, Chapter Six discusses the conclusion, contribution, and future work.

Chapter Two: Literature Review

2.1 Introduction

This chapter discusses the state of the art about the distributional semantics and explains about the Amharic language, its writing system, and its characteristics. It defines the semantic vocabulary and states more about distributional semantics in detail. It also discusses the models used to embed the words across a vector space, which includes latent semantic analysis, explicit semantic analysis, non-negative matrix factorization, Word2Vec, Doc2Vec, and paragraph vector. It discusses the application areas where semantic vocabulary can be applied like information retrieval using query expansion and it informs that the documents must first be indexed to be searched later using query expansion. The chapter also explains the semantic similarity measurements that calculates how much the words are far apart or close in meaning either based on angular orientation like cosine and based on the magnitude of word's distances. To the end, it discusses the performance evaluation metrics and the summary of the contents presented.

2.2 Amharic Language

Amharic is the official and working language of Ethiopia which has dominant speakers. It is a Semitic language that is rooted from the Ancient language called Geez (Ge'ez) which was the official language of Ethiopia before Amharic [16]. Since the 13th-century Amharic has been the language of courts, language of trade and everyday communications, the military, and the Ethiopian Orthodox Tewahedo Church and remains the official language of the Country. It is a morphologically rich and resource-limited language and it has no free data set available for research works. It is the second-most spoken Semitic language in the world, after Arabic and has four dialectical variations like Gojam, Gonder, Wollo, and Menz which is spoken little different from the popular Amharic dialect. Even if it is the official and most dominant language inside Ethiopia it has also many speakers outside the country like Israel. It is a morphologically complex language that one word can be interpreted and expressed in many ways. It is different from other languages in which it has its own Alphabets and syllabic patterns. The part of speech order is differing from English having of subject-object-verb agreement (SOV). As English has 26 alphabets, Amharic language has 34 common alphabets from ʋ to ፕ including ፘ.

2.2.1 Amharic Writing System

Amharic is written with a version of the Ge'ez script known as Fidel. It has 34 alphabets with 7 vowels which have more than 238 different characters. There are seven vowels in Amharic which is used to make readable the consonants by inserting anywhere in the text. The seven vowels include a, e, o, u, ə, i, and ä. e and ä are used in the first order of the consonant lists, u is used in the second-order, i is used in the third order, a is used on the fourth-order, ə and ï are used in the sixth order where the last one, o is used at the seven orders of the consonants. For example, säbärä to mean he broke, säbabärä to mean he shopped down, säbärächïgn or säbärächəgn to mean she broke me, etc. is written with a consonant with vowel combinations. It is more clearly depicted in the following table 2.1.

Table 2.1: Amharic Alphabet

1st order ä/e	2nd order u	3rd order i	4th order a	5th order ē	6th order ï/ə	7th order o	Combination with was
ሀ (hä)	ሁ (hu)	ሂ (hi)	ሃ ha	ሄ hē	ህ hï / hə	ሆ ho	
ለ (lä)	ሉ (lu)	ሊ (li)	ላ la	ሌ lē	ል lï / lə	ሎ lo	ሏ lwa
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
ፐ (pä)	ፑ (pu)	ፒ (pi)	ፓ pa	ፔ pē	ፕ p / pi	ፖ po	ፕ pe

2.2.2 Characteristics of Amharic Language

The Amharic language is different from English due to no means of capitalization, Subject Object Verb (SVO) patterns of speech order, has its own punctuation marks in common with a question mark and can use geez numbering system. It has features like complex morphology and spelling inconsistency.

Amharic has variable features and its nature leads difficulty for natural language processing and understanding. For example, different spelling inconsistencies and homophone characters like ጸሀይ(tsähay), ጸሓይ(tsähay), ጸኀይ(tsähay), ጸኃይ(tsähay), and ፀሀይ(tsähay) to mean sun makes the machine difficult to decide. It is also morphologically complex that one word may have many morphemes. E.g. የጤና(yätēna), ለጤና(lätēna), ከጤና(kätēna), ጤናማ(tēnama), ጤንነት(tēnənät), ጤናችን(tēnachən), etc for the stemmed word ጤና(tēna) to mean health.

Like other languages, Amharic has an abbreviation. We can write the long strings in its standard short form. The short form can be shortened using period or forward slash in the middle of the

words. For example, አ/አ or አ.አ to mean Addis Ababa (Addis Abäba) and ዓ/ም or ዓ.ም to mean amete mhret (amätä məhərätə) must be normalized into long-form.

The ambiguity and vagueness of the language are more complex than English. One single word may be interpreted as too many meanings. There is the canonical and common writing system of the languages as it is shown in the following table [15]. The different writing system of compound words like the word palace to mean betemengst can be written in two ways ቤተ-መንግስት or ቤተ መንግስት. This is clearly shown in Annex B of the last page.

Table 2.2: Amharic Spell Variation

Canonical Amharic	**Common Amharic**	**Possible but improbable**
ዓለም	አለም	ዐለም, ኣለም
ፀሐይ	ጸሃይ, ፀሃይ, ጸሐይ	ጸሀይ, ጸሓይ, ጸኀይ, ጸኃይ, ጸኻይ, ፀሀይ, ፀሓይ, ፀኀይ, ፀኃይ
ኃይል	ኀይል, ኃይል, ኀይል, ኀይል, ኃይል ሀይል, ሀይል, ሀይል, ሃይል, ኀይል ሐይል, ሐይል, ሐይል	ሐይል, ሓይል, ሓይል, ሓይል, ኻይል, ኻይል
አዲስ አበባ	አዲስ አበባ	ዓዲሥ ዐበባ (not acceptable)
ኢትዮጵያ	ኢትዮጵያ	ዒትዮጵያ

Such type of writing system increases the difficulty of the machine to understand than the speaker and the language experts.

2.3 Overview of Vocabulary

Semantic Vocabulary: - It is the body of similar words used in a language and known to an individual person. It is all about similar words in a language or a special set of words we are trying to learn. Vocabulary is the knowledge of words and word meanings. As Steven Stahl (2005) put it, vocabulary knowledge is knowledge and the knowledge of a word not only implies a definition but also implies how that word fits into the world. Vocabulary knowledge is not something that can ever be fully mastered, but it is something that expands and deepens over the course of a lifetime. Instruction in vocabulary involves far more than looking up words in a dictionary and using the words in a sentence. Most of the time vocabulary is acquired incidentally through indirect exposure to words and intentionally through explicit instruction but also implicitly constructs in specific words and word-learning strategies [17]. So, this work is mainly focused on the implicit construction of semantically related vocabulary from the

Amharic document. It is, used in reference to verbal expression, calls attention only to the extent or variety of the writer's or speaker's stock of words or to the sources from which such stock is derived. It is a collection of semantically related words.

Dictionary: - It is the usual term for a book which gives not only the words that belong to a language but also their meanings, their accepted spelling, their pronunciation, etymology, and the like; as Webster's new international dictionary of the English language, Oxford English dictionary and as of our Amharic dictionary. It is also the general term applied to a book that embodies an alphabetical list of names with explanatory information or that presents an alphabetical list of terms with their synonyms. It may be the same as in one or another way to the glossary term of the books with the definition of each term.

Thesaurus**:** It presents words as word families, listing their synonyms without explaining their meanings or usage which can be alphabetically or conceptually. This is like vocabulary but, orders may be a must when too many lists of words are available. List of words grouped together according to the similarity of meanings or synonyms and sometimes antonyms.

2.4 Distributional Semantics

As it is stated in chapter one distributional semantics is one approach of semantics in which the meaning is extracted from word contexts and its distributions across a vector space. It is a theory of meaning which is computationally implementable and very good at modeling what humans do when they make similarity judgment and it models computational representation of word meaning from the patterns of co-occurrence of words in a given text which can provide reliable estimates of meaning relatedness for many semantic tasks requiring them. It extracts meaning information exclusively from a text which states words that occur in similar contexts are semantically similar [18]. It is statistical and data-driven and focuses on aspects of meaning related to descriptive content [19]. It is the way of relating linguistic entities like words, terms, phrases, sentences, and documents to each other based on their distributional properties in a corpus that requires no external resources [13]. Distributional semantics is the one that develops and studies theories and methods for quantifying and categorizing semantic similarities between linguistic items based on their distributional properties in a large corpus.

Dimension Reduction

It is the process of reducing the size of the original corpus under consideration by obtaining a set of principal variables and it is about changing high dimensional vector into low dimensional meaning representations. There are approaches for reducing the dimension of the original input matrix-like SVD which can be used in combination with LSA language modeling and Random Indexing (RI). Singular value decomposition is clearly defined in the LSA of section [2.5.1]. Random indexing is a computational framework for distributional semantics, based on the insight that very-high-dimensional vector space model implementations are impractical, that models need not grow in dimensionality when new items are encountered. The high-dimensional model can be projected into a space of lower dimensionality without compromising distance metrics if the resulting dimensions are chosen appropriately. The Word2Vec algorithm is used as a dimension reduction up to a certain size as the user wants to embed and the dimension size between 200 and 300 is advisable. We have reduced the embedding size into 300 dimensions which brought us a better word analogy. The advantage of dimensional reduction is to be computationally flexible and to have a one-time operation.

2.5 Distributional Semantics Models

Distributional semantic models (DSMs) approximate word meanings with vectors recording their patterns of co-occurrence with corpus contexts [20]. Various statistical methods have been applied to extract the semantic information from a given corpus and many computational models have been proposed.

2.5.1 Latent Semantic Analysis (LSA)

It is a technique in natural language processing distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. It is defined as a mathematical method that tries to bring out latent (hidden concept) relationships between words or phrases within a corpus that focuses on co-occurrence of terms in a document and creating a term-document matrix which maps terms (documents) to a vector space of reduced dimensionality [13].

According to a distributional hypothesis, LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph in which rows represent unique words and columns represent each paragraph is constructed from a large piece of text and singular value decomposition (SVD) is used to reduce the number of rows

while preserving the similarity structure among columns. Then the meaning of words is compared using cosine similarity measure to show how much words are close to each other or far apart in meaning. Atypical example of a term-document matrix using LSA is TF-IDF (Term Frequency-Inverse Document Frequency) that the weight of term frequency has a significant effect on the relative importance of word meaning.

According to Martin *et al.* [21] LSA is an approach using linear algebra for effective yet automated information retrieval that uses Vector Space Model (VSM) to handle a text retrieval from a large heterogeneous document. It was originally designed to improve the effectiveness of information-retrieval methods by performing retrieval based on the derived semantic content of words in a query as opposed to performing direct word matching and it is a statistical model of word usage that permits comparisons of the semantic similarity between textual information [22]. It works by extending the vector-based approach using Singular Value Decomposition (SVD) to reconfigure the data having a set of underlying latent variables which spans the meanings that can be expressed in a particular language [23]. The core idea behind LSA is it represents words across a multi-dimensional space based on the count of terms within a certain document.

Probabilistic latent semantic analysis is a novel statistical technique for the analysis of co-occurrence data, which has applications in natural language processing. Compared to standard latent semantic analysis which stems from linear algebra and downsizes the occurrence tables using singular value decomposition, probabilistic latent semantic analysis is based on a mixture decomposition derived from a latent class model [24]. Its main goal is to model co-occurrence information under a probabilistic framework in order to discover the underlying semantic structure of the given data.

2.5.2 Explicit Semantic Analysis (ESA)

It is a variation of semantic analysis based on explicit concepts and it contrasts with latent semantic analysis (LSA) because the use of a knowledge base makes it possible to assign human-readable labels to the concepts that make up the vector space. In ESA, a word is represented as a column vector in the TF-IDF matrix of the text corpus and a document which is a string of words can be represented as the centroid of the vectors representing its words. It is a method of representing the meaning of text fragments that mimics the way human beings think about words, their meanings and the relationships between words and phrases which is used for finding the semantic relatedness between text fragments [25].

The difference between ESA and LSA is the concepts are already known and labeled in ESA, whereas in LSA the concepts are latent and they are undefined and need to be discovered. The explicit semantic analysis uses the knowledge base like Wikipedia to create an inverted index that maps words to contents and then operates over this vector representation of words, where each word is now a vector of titles with 0 and 1. Whereas LSA uses Singular Value Decomposition principle to project the word-doc matrix into a lower-ranked space such that dot product of word-doc vector representation of words that do not co-occur with each other in any document, but they co-occur with a similar set of words. In contrast to LSA, ESA uses a knowledge base to make the vector space conceptual, so that ESA is explicit. It is a novel method that represents the meaning of texts in a high dimensional space of concepts derived from the knowledge base [26], [27].

Explicit Semantic Analysis (ESA) has been recently proposed as an approach to computing semantic relatedness between words and texts that has a natural application in information retrieval, showing the potential to alleviate the vocabulary mismatch problem inherent in standard BOW and the model has been also recently extended to cross-lingual retrieval settings, which can be considered as an extreme case of the vocabulary mismatch problem [28]. Generally, ESA is one of the languages modeling computational techniques in which it uses a prior knowledge base like Wikipedia for meaning or knowledge representations.

2.5.3 Non-Negative Matrix Factorization (NNMF)

Non-Negative Matrix factorization is a group of algorithms in multivariate analysis and linear algebra where a matrix V is usually factorized into two matrices W and H, with the property that all three matrices have no negative elements. The non-negativity value makes the resulting matrices easier to inspect and the problem it has is not exactly solvable in general but can be approximated numerically.

According to Yongxia *et al*. [29] the aim of NNMF is to classify a given N-dimensional document into a set of separated low dimensional texts to realize text feature selection and text feature vector representation. It is mainly used for topic modeling, text classification, and text summarizations. This shows that the classification of the original document into N low dimensional space can reduce the dimension of vectors and computation overhead of similarity calculation between text feature vectors. It was proposed to solve the problem of high computational overhead and low classification efficiency of the KNN algorithm. In distributional semantics, it is used for dimension reduction like SVD. It was used by Baroni

and Zamparelli (2010) which is a less commonly adopted method, but it has also been shown to be an effective dimensionality reduction technique for distributional semantics (Dinu and Lapata, 2010).

2.5.4 Word2Vec

Using unsupervised learning models like Word2Vec, the distributed vector representation of words ignores the traditional bag of words which has the values only one-hot encoded vector either zero or one. This affects the semantic extraction by which values approximate to 1 like 0.985 will not be considered, because of the values is set either zero or one. Due to this, the unsupervised learnings are the best semantic vector representations for achieving a certain research goal.

Word2Vec was developed by Tomas Mikolov, *et al.* at Google (2013) as a response to make the neural-network-based training of the embedding more efficient and since then has become the de facto standard for developing pre-trained word embedding. It is an improved and extended model of the previously known approaches for efficient estimation of semantic word representations in vector spaces. The Word2Vec takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words in which the resulting word vector file can be used as features in many natural language processing applications including vocabulary construction [30]. For example, vector operations [vector('King') - vector('Male') + vector('Female') is close to vector('Queen')] capturing the analogy King is to Queen as Male is to Female. Word2Vec has two different learning architectures called the continues bag of words (CBOW) model that learns the embedding by predicting the current word based on its context and the continuous skip-gram model which learns by predicting the surrounding words given a current word.

2.5.4.1 Continuous Bag of Words (CBOW)

The idea behind CBOW architecture is meaning can be inferred from a word surrounding another word and specifically the notion about it is by providing the surrounding words the algorithm predicts the target or center words. The argument value for CBOW 0. It is better for a small corpus size and very fast compared to skip-gram. As it is shown in figure 2.1 if the model is learned in CBOW it predicts the ‘mat’ given the context words ‘the cat sits on the’. Word2Vec implements binary classification for both SG and CBOW to discriminates exactly the target words and context words.

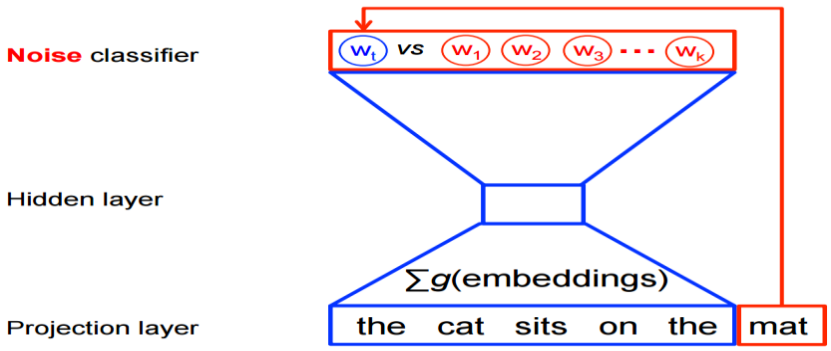

Figure 2.1: CBOW to discriminate the target words *[The image was adopted from tensorflow.org]*

$$J_{\text{NEG}} = \log Q_\theta(D=1|w_t,h) + k \underset{\tilde{w}\sim P_{\text{noise}}}{\mathbb{E}} [\log Q_\theta(D=0|\tilde{w},h)] \quad (1)$$

The objective of the binary classification is to maximize the probability likelihood of the target or the context words to be predicted, where $Q_\theta(D=1|w_t,h)$ is the binary logistic regression probability under the model of seeing the word w in the context h in the dataset D, calculated in terms of the learned embedding vectors θ. In practice, we approximate the expectation by drawing k contrastive words from the noise distribution. When the model assigns the high probability value it becomes the target word else it is noisy words. K number of samples is selected from V vocabulary size when there is a large document size to make it fast training.

2.5.4.2 Skip Gram (SG)

This architecture learns the vectors in the way in which word contexts are represented across a vector space that can predict the surrounding words given the target word. The argument value for skip-gram is 1. According to figure 2.1, the skip-gram predicts the context words like 'the', 'cat',' sits',' on' 'the' given 'mat' which is the target word.

Word2Vec is unsupervised learning that is capable of utilizing unlabeled data to convert a word into its vector representation to find the semantic relationship between words by counting their distance and the vector representation can catch the semantic similarity between words by which the similarity feature is used for text classification [31]. It is an open-source tool that is proposed by (Mikolov *et al.* 2013) which can map word to real vector and is considered as a perfect estimation of word representations in vector space which has the main significance of large improvements in accuracy at much lower computational cost, so meaning of words can be represented better [32]. It can also be used in combination with other semantic models like LSA for better accuracy.

The performance of Word2Vec is hugely depended on the training settings like CBOW, skip-gram, factor size, windows size, and frequency threshold. Language features are extracted using Word2Vec by summing all the vector representation of the tokens that exist in the document and dividing it by the total number of vectors [31]. Generally, Word2Vec is the powerful and more accurate meaning representation of distributional semantics and the distributional semantic models are an extension of neural networks.

2.5.5 Doc2Vec

Doc2vec is an extension of Word2Vec that learns to correlate labels and words, rather than words with other words so, it is used to associating arbitrary documents with labels. It is used for the distributional representation of documents which is based on the distributional hypothesis that words in documents occurring in similar contexts tend to have similar meanings. Proposed by Le and Mikolov (2014) to extend the learning of embedding from words to word sequences.

The author Jey Han Lau and Timothy Baldwin [33] explained that Doc2vec was proposed in two forms: DBOW and DMPV by which DBOW is a simpler model and ignores word order, while DMPV is a more complex model with more parameters and DBOW is better than CBOW and the findings reveal that DBOW, despite being the simpler model, is superior to DMPV. When trained over large external corpora, with pre-trained word embedding and hyper-parameter tuning, they found that doc2vec performs very strongly compared to both a simple word embedding averaging and n-gram baseline, as well as two state-of-the-art document embedding approaches. The doc2vec performs particularly strongly over longer documents. Doc2vec (Paragraph2Vec) modifies the Word2Vec model into unsupervised learning of continuous representations for larger blocks of text, such as sentences, paragraphs or entire documents, meaning Doc2vec learns to correlate labels and words rather than words with other words [34].

Doc2vec learns to capture not just individual words but the entire sentence and paragraph. It uses both bag-of-words and word n-grams models producing the new state of the art results for many natural language processing applications including vocabulary construction. It requires the number of features to be returned (length of the vector), the size of the window that captures the neighborhood and the minimum frequency of words to be considered into the model in which the values of these parameters depend on the corpus for better accuracy and performances [35].

2.5.6 Paragraph Vector

Paragraph vector is an unsupervised distributional semantics algorithm that learns a meaning representation from variable-length pieces of texts, such as sentences, paragraphs, and documents that represent each document by a dense vector that is trained to predict words in the corpus. It outperforms a bag-of-words model for text representations [36] and learns continuously distributed vector representations for pieces of texts that can be applied for any variable-length texts. The concept behind the paragraph vector is the vector representation can be trained to be useful for predicting words in each paragraph. After the model is trained, the word vectors are mapped into a vector space such that semantically similar words have similar vector representations.

Author Ruqing Zhang *et al.* [37] introduced a generative paragraph vector, which can be viewed as a probabilistic extension of the distributed bag of words version of the paragraph vector with a complete generative process with the ability to infer the distributed representations for unseen texts. They could further incorporate text labels into the model and turn it into a supervised version, namely a supervised generative paragraph vector.

Vector space models (VSMs) represent (embed) words in a continuous vector space where semantically similar words are mapped to nearby points ('are embedded nearby each other') and the methods depend in some way or another on the distributional hypothesis, which states that words that appear in the same contexts share semantic meaning. The word-space model (WSM) is a computational model of meaning to represent the similarity between words/text and it derives the meaning of words by plotting the words in an N-dimensional geometric space.

2.6 Information Retrieval Using Semantic Vocabulary

Nowadays the need access to information retrieval becomes more likely increasing due to its availability and a new advance in technology and possible scanning of the internet. Information can be reached via social media like Facebook, YouTube, messenger or public media such as radio, television, and video conferencing or Database and indexed resources. Our work is focused on providing information search and retrieval through indexed text documents with expanded query terms using semantic vocabulary to increase search relevance with high document ranks. This shows that our system can be used for enhancing information search and retrieval.

The term information retrieval [38] can be applied for both unstructured and semi-structured text data as opposed to database files which involve filtering of a large amount of data based on the contents it is containing. Even though information retrieval is basically applied for text data, but it includes multi-media files like audio, video, image, and graphics in which it enables the users to satisfy his or her information need. It is the way in which users gain relevant results for a given query from a data source which can be a database or indexed documents.

Basically, information retrieval is categorized into two parts. The first one is based on an unranked set of retrieval which returns a set of relevant documents for the user's query without prior relevance order or ranking. This may be tedious and tiresome while looking for search results. The second one is based on a ranked retrieval system that the relevant documents are retrieved in descending order. It returns a ranked list of relevant results instead of a set of unordered lists for example, google applies such kinds of techniques. This enables the users to quickly access the set of relevant documents and saves time for the users while seeking the most relevant documents during searching. The user's happiness is more warranted while using the ranked retrieval. We have implemented a ranked retrieval system with semantic vocabulary as query expansion.

Query expansion is necessary to improve the information relevancy which can guarantee the need of the users while seeking relevant documents. Instead of retrieving the documents based on a single query it is better to extend the query terms to its corresponding semantic words. The query can be extended using the semantic thesaurus, WordNet or semantic vocabulary. These expansion resources can be constructed either automatically or manually from large corpora.

Document indexing and searching are one side of the same coin. A set of documents are preprocessed and indexed for later search and retrieval. Indexed data is structured and easy to handle and retrieve. It is briefly discussed in the following sections.

2.6.1 Document Indexing

In the concept of all information retrieval systems, the central idea of indexing and searching documents for efficient result lookup is common nowadays which enables rapid searching and processing. Indexing and searching are dependent on each other. They are like the two sides of the same coin. Once the document is indexed it must be imported and be searched. Whoosh is

a pure python library that is used for indexing and searching text documents. It is not a full-fledged application in which users cannot directly download and install, but it is an API. Users can install it via python command prompt and import using any python editor to index or search text documents.

As Apache Lucene is a common implementation for document indexing and searching in java, whoosh is common for full python implementation of text document indexing and searching. Using whoosh does not require Java made indexing tools to python like pylucene. It is used directly with python programing language. Hence our work is implemented in python, we preferred to use whoosh. It is fast when indexing and basically used for increasing document score rankings, especially when using query expansions, the result for document ranking becomes more reasonably increasing.

Features of Whoosh

- ✓ It is a search engine entirely written in python which is more likely the same as Apache Lucene has written fully in Java.
- ✓ It is quite flexible as compared to other searching and indexing engines.
- ✓ Even it has a little limitation like accurate results may not be recorded as it is expected for other languages other than English, it has more features like regex searching, substring searching, rank searching and sorting
- ✓ It is an offline search engine as opposed to others like scout which is server-based on SQLite database
- ✓ Using whoosh, we can search even keywords and date or time, because it allows as to index parts of strings as field values.
- ✓ It can add documents to the reverse index by examining the corpora.
- ✓ It is also possible to reference the URL of the file.
- ✓ Suggestion to mistyped words and spell correction.
- ✓ It highlights the text matching in the document documents.
- ✓ Built-in Analysis and elimination of stop lists for the English language.
- ✓ It is also possible for n-gram searching.

Whoosh is fast, easy to use and implement, has high performance, fits with full-text search with a high probability of gaining results for a query and advisable for fewer duty tasks like offline data searches.

Before indexing of the documents, the schema must be defined. Schema is the high-level structure of data in which the field elements are mapped and defines which kind of field is going to be indexed. For example, it defines field elements like keyword, ID, path, DATETIME, text, and title. So, after the index is written into the index writer all the information becomes commit to writing and permanently saved in the disk.

2.6.2 Document Searching

Searching is the way of looking for relevant documents by feeding the query into the system which makes the resource available. Documents can be searched from structured sources like a database or semi-structured sources like indexed system. Users can search for information from unstructured sources like multi-media files e.g. audios and videos. Our work is focused on retrieving information from an indexed document by providing queries expanded from the semantic vocabulary which increases search relevance.

There are three main Boolean query searches called OrGroup, AndGroup and Not. Using OrGroup best results can be retrieved. For example, let the query be Africa Olympic the document containing either Africa or Olympic or containing both words can be retrieved. The notion behind ORGroup Boolean search is if the query terms match in any one of the field elements the relevant documents will be retrieved. In the case of AndGroup it searches the documents exactly containing Africa and Olympic will be retrieved. In this case, results will be less relevant and retrieved. It is mostly used in the case of authentications like password and username matching which is not advisable for information retrieval. Using Not Boolean searching documents containing Africa but not Olympic will be retrieved. After matching for query terms is found the rank of relevant documents, the path of the file, the term scoring, the title, and the text data is retrieved for the users.

During searching, users can provide the query to the system and the terms are expanded with the semantic vocabulary and finally, the matched results will be returned for the users.

The relevance of documents is query-based in which the similarity of the query with the matched documents can be calculated and ranked by which Google applies such types of techniques. There are three common term scoring algorithms called **frequency**, **TF-IDF**, and

BM25F. So, the relevant documents are retrieved according to the ranking function by expanding the terms to semantic vocabulary.

1. Frequency

It returns the relevant documents based on the count of terms found in the document. This scoring method is basically used when no weighting and normalization required.

2. TF-IDF

TF-IDF is the simplest and the most common intuitive term scoring function containing two basic parts called term frequency (TF) and inverse document frequency (IDF). The first one is defined as the number of query terms how often occurring in each document and IDF dictates that the frequency of terms that are occurring in a document. It defines the numerical values that show how a certain query term is important to a document. It is becoming more popular term weighting function mainly used in a text-based recommender system, text mining and user modeling. It is interpreted separately like:

TF(t, d)=(number of times term t appears in a document)/(total number of terms in the document)

IDF(t, D)=log(total number of documents/number of documents term t appears). To discriminate the bias towards longer document size TF(t, d) is added and multiplied by 1/2.

$$TF(t,d)=0.5+0.5*\frac{ft,d}{Max\{ft',d:t'\in d\}} \quad (1)$$

$$IDF(t, D)=\log\frac{N}{|\{d\in D:t\in d\}|}, \quad (2)$$

where N is the total number of documents and $|\{d \in D: t \in d\}|$ is the number of documents t appears in. Finally, the TF-IDF is calculated as TF-IDF(t,d,D)=TF(t,d)*IDF(t,D) and the relevant documents are returned for the respective query terms using TF-IDF values in descending order.

3. BM25F

It stands for best matching version 25F which is a ranking function used by the information retrieval system. It is used to rank a document according to their relevance for given query terms and implements state-of-the-art TF-IDF ranking function.

$$BM25F(D,Q)=\sum_{i=1}^{N} IDF(qi) * \frac{f(qi,D)*(k1+1)}{f(qi,D)+k1*(1-b+b*\frac{|D|}{Avgd1})} \quad (3)$$

,where f(qi, D) is the query term frequency in document D, |D| is the length of document D in words, Avgd1 is the average document length in the text collection, k1 and b are free parameters as k1∈ [1.2,2.0] and b=0.75 and IDF is the inverse document frequency computed as IDF(qi)=log$\frac{N-n(qi)+0.5}{n(qi)+0.5}$ (4)

The minimum requirement to score the searching relevance is the searcher, field number, text document number, and weighting. Using BM25F term scoring is smoothly weighted and the IDF is taking the length of each document into consideration in which the same terms appearing in short document will score higher. It supports the scoring of terms across multiple weighted fields like path, ID, title, body, text, and URL.

Semantic vocabulary can be integrated into an information retrieval system and the users can search the most relevant documents for their queries. Too many numbers of documents can be retrieved by expanding the queries into the semantic vocabulary. Expanding with the semantic vocabulary can increase the recall of searching.

2.7 Similarity Measurement

Semantic similarity is a metric defined over a set of documents or terms, where the semantic distance between words is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical. The term semantic similarity is often confused with semantic relatedness. Semantic relatedness includes any relation between two terms, while semantic similarity only includes **is-a** relation. Our work is focused on semantic relatedness, especially for contextual meaning. Semantic similarity measures are techniques used to estimate the strength of the semantic relationship between units of documents, concepts or instances, through a numerical description obtained according to the comparison of information supporting their meaning or describing their nature.

There are three types of similarities known as string-based, corpus-based and knowledge-based, where the first one is operating on string sequences and character composition and the second one is determining the similarity between words according to information gained from large corpora. The third determines the degree of similarity between words using information derived from semantic networks. The followings are the most common and known similarity measures.

2.7.1 Cosine Similarity Measurement

Semantic similarity between words is usually measured by the cosine similarity of their corresponding vectors. This metric finds the normalized dot product of the two inputs then calculating the angle between input vectors. It is about a judgment of orientation and not magnitude. Two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. It is particularly used in positive space, where the outcome is neatly bounded in [0, 1]. It is the popular similarity measure because of very efficient to evaluate, especially for sparse vectors [39].

Given two vectors of attributes, A and B, the cosine similarity, cos(θ), is represented using a dot product and magnitude as Similarity$(\cos\theta)= \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{N} AiBi}{\sqrt{\sum_{i=1}^{N} Ai^2}\sqrt{\sum_{i=1}^{N} Bi^2}}$ (5)

, where Ai and Bi are components of vector A and B respectively. Cosine similarity distance is calculated as Dc (A, B) =1-Similarity (Cosθ).

According to the author Anna Huang [40] when documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors and this is quantified as the cosine of the angle between vectors, that can be applied for text documents, such as information retrieval (R. B. Yates and B. R 1999, modern information retrieval). Each dimension represents a term with its weight in the document.

2.7.2 Jaccard Similarity Measurement

It's a measure of similarity for the two sets of data, with a range from 0% to 100%. The higher the percentage, the more similar the two text strings and it is easy to interpret. It is extremely sensitive to small sample sizes and may give erroneous results, especially with very small samples or data sets. The Jaccard coefficient measures the similarity between finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets.

It is computed as $J(X, Y)= \frac{\sum_{i=1}^{N} Min(Xi,Yi)}{\sum_{i=1}^{N} Max(Xi,Yi)}$ (6)

Jaccard distance as $d_j(X,Y)=1-J(X,Y)$. (7)

, where $X=(X_1, X_2, X_3,\ldots, X_n)$ and $Y=(Y_1, Y_2, Y_3,\ldots, Y_n)$ are two vectors with all real X_i, Y_i,

The Jaccard algorithm measures the similarity between finite numbers of documents sets like to find out the matching between training and testing set for opinion mining of roman texts based on the clustering system [41]. This is not the best semantic measure when using distribution representation of words, because the vectors are not learned based on the number of terms that the document contains but based on its surrounding of words.

2.7.3 Euclidean Distance Similarity Measurement

The Euclidean distance or Euclidean metric is the ordinary straight-line distance between two points in Euclidean space. Using this distance, the Euclidean space becomes a metric space and is referred to as the Pythagorean metric. The Euclidean distance between point's p and q is the length of the line segment connecting them ($\overline{pq}$). In Cartesian coordinates, if p = (p1, p2,..., pn) and q = (q1, q2,..., qn) are two points in Euclidean n-space, then the distance (d) from p to q, or from q to p is given by the Pythagorean formula:

$$d(p, q) = d(q, p) = \sqrt{(q1 - p1)^2 + (q2 - p2)^2 + \ldots + (qn - pn)^2}$$

$$= \sqrt{\sum_{i=1}^{N} (qi - pi)^2} \quad (8)$$

According to the author Anna Huang [40] Euclidean distance is a standard metric for geometrical problems and it is the ordinary distance between two points which can be easily measured with a ruler in two- or three-dimensional space. It is widely used in clustering text clustering. It is also the default distance measure used with the K-means algorithm.

2.7.4 Gensim Similarity Measurement

The two basic semantic measures that the python gensim library provides and the Word2Vec implements are most_similar() and most_similar_cosmul(). Both metrics implement the concept of cosine metric which operates using the word's orientation or angular distances rather than their magnitude with additional smoothing and averaging to increase the cosine distance values. The most_similar() method calculates the mean projection of a weighted vector between the query word and the words in the vector space. The most_similar_cosmul() implements the argmax(cos(A, B)) with an additional parameter called epsilon.

2.8 Performance Evaluation Metrics

The performance of the system is tested by considering two basic folds.

2.8.1 Evaluating Unsupervised Learning

The quality of unsupervised learning is determined by the end goal after it is learned. Word2Vec is an instance of unsupervised learning that performance and accuracy cannot be directly measured using precision and recall. But it can be measured using either extrinsic or intrinsic methods which are discussed as follows.

Extrinsic: - it is an indirectly measuring technique that can increase the performance of the system like using parts of speech tagging and named entity instead of using the entire documents. It may be better to learn the model using specific parts of speech like nouns or verbs.

Intrinsic: - It implies measuring the performance of the system using methods like word's syntactic or semantic relations. Syntactic relation is based on the grammatical structure and certain rules whereas semantic relation focusing on the meaning of words. It is more discussed based on the following measures.

2.8.1.1 Analogical Measure

As it is proposed by Miklove *et al* 2013 we can measure the Word2Vec accuracy using analogy semantic relations. The notion behind analogy is to find a word m for a given word n so that m: n best resembles as a sample relationship w: w' or in a simple expression if a is to b as c is to d.

2.8.1.2 Relatedness

It defines cosine values between a set of words approaching 1 is highly related. For example, the cosine similarity of city and united is 0.998855307698038 which means City is highly related to the term United than other words else in the vocabulary. So generally, Word2Vec by itself can be evaluated using intrinsic evaluation. But to make it visible and clearer how much our semantic vocabulary is used, we have implemented in the area of information retrieval by expanding the original query into a set of reformulated similar words.

2.8.2 Evaluating the Semantic Vocabulary with Information Retrieval System with Query Expansion.

It is possible to evaluate the recall and precision by integrating into an information retrieval system. The metrics are primarily based on four concepts of (True Positive, True Negative, False Positive, False Negative) which can be abbreviated as (TP, TN, FP, FN). It is emphasized

on increasing the number of relevant documents to be retrieved when using the semantic vocabulary as query expansion.

Precision: - It is the fraction of the retrieved documents which is relevant.

$$P = \frac{TP}{TP + FP} \text{ or, } \frac{relevant \cap retrieved}{retrieved} \quad (9)$$

Recall: - It is the fraction of the retrieved documents which has been retrieved.

$$R = \frac{TP}{TP + FN} \text{ or, } \frac{relevant \cap retrieved}{relevant} \quad (10)$$

2.9 Summary

We have discussed the insight about Amharic language and the application of distributional semantics with respect to the models and some similarity measure of meaning representations. Latent semantic analysis with SVD is suitable for linear meaning understanding and Word2Vec is advisable for distributional representation. Even if the Jaccard similarity measure is not common and even has the same advantages as cosine similarity cosine is advisable for distributional distance calculation. Explicit semantic analysis is commonly used for a structured knowledge base for meaning inferring. Doc2Vec and Paragraph vectors are commonly used for term-document representations.

Before representing the semantics of the words, the input corpus must be properly tokenized and preprocessed because representing the meaning of words in its raw text leads to an error. The dimension of the text is minimized using a dimension reduction algorithm for better accuracy and reducing computation time. We have examined each technique for how to build lexical resources automatically from a text corpus and its use for enhancing the information retrieval system.

The reason why we preferred to apply the semantic vocabulary for information retrieval is to make it available a result for searching query terms and to increase recall. Instead of searching with only in the unexpanded query, searching with expansion brings more relevant results with a high probability of gaining a matched document.

Chapter Three: Related Work

3.1 Introduction

This chapter discusses problems and solutions proposed for semantic lexical resources construction automatically or semi-automatically from a corpus using unsupervised techniques. We have focused to review research works done using corpus-based because it is possible to construct semantic vocabulary automatically from Amharic corpus. This section also informs how and where the corpus is collected. It also discusses the application domain where lexical resources can be applied.

Although many approaches have been proposed to identify the semantic relatedness of terms, it is still very hard to identify which is the best method to use and, in this part, many works regarding thesaurus and vocabulary construction using distributional semantics is reviewed. In addition, to what extent the authors achieved their works and the application program interface (API) used during the implementation and experimentation of the research works is assessed.

There are many works done using distributional semantics like automatic construction and expansion of vocabularies and thesaurus for both local and Non-Ethiopian languages like English, Japanese, Swedish, French, and Amharic.

3.2 Information Retrieval

Once a lexical resource is built it is used as expanding the user's query for information retrieval and it can better improve the user's satisfaction while looking for relevant documents. But the nature of the semantics, the size and the analogy of terms affect the retrieval performances. The resources constructed using distributional semantics approach is better, especially for large data size. Lexical resources using a distributional semantic approach is used for information retrieval and extraction for non-Ethiopian languages so far. There is an attempt done for automatic thesaurus construction for Amharic using the traditional count of words approach using LSA with SVD. It doesn't consider contextual similar words. The following section talks about lexical resource construction for Ethiopian and non-Ethiopian languages.

3.3 Vocabulary Construction for Non-Ethiopian Languages

Non-Ethiopian languages are languages speaking outside the country. It is the language that the natives outside Ethiopia speak. It includes English, French, Spanish, Nepal, Kiswahili. Chinese, Swedish, etc. Many of the people speak English as a second language in addition to their mother tongue. Because money of the curriculum is drafted in English and people all over

the world become communicated using English. It is an international language that many of the research papers and projects are written in English. There is an attempt for lexical resources construction for foreign languages like a thesaurus, WordNet, and semantic vocabulary for information retrieval created either manually or automatically from a large text corpus.

3.3.1 Thesaurus Construction for English

Curran and Moens [42] conducted research on how to improve automatic thesaurus construction using different techniques in which the document was collected from the British National Corpus (BNC), containing 114 million words and 6.2 million sentences. The research was focusing on both semantic and syntactic patterns of a text corpus. They articulated that the system can extract related terms directly by recognizing linguistic patterns that link synonyms and hyponyms (Hearst, 1992; Caraballo, 1999) relations. The context relation was defined as a tuple of (w, r, w') where w is thesaurus term, r is the relation and w' is words found in the corpus. After tagging and chunking using Naïve Bayes POS, the noun phrase separated by preposition can be concatenated and finally the relation extraction algorithm is applied. Then the semantic relatedness of the words with its relation is compared with the words from the WordNet and terms that are close to semantic distance can be considered as a thesaurus. The authors implemented Jaccard similarity measure for semantic distance calculations that showed the best performance with high accuracy. To evaluate the system, they considered 70 thesaurus terms and the words are randomly selected from the WordNet. Then the contextual similar meaning of words that are randomly selected from the WordNet and terms from the corpus was compared and many semantically related thesauri were constructed.

Claveau and Kijak [43] proposed distributional thesauri for information retrieval. The corpus was collected from AQUAINT-2 composed of articles in English containing a total of 380 million of words. For a given input word these thesauri identify semantically similar words based on the assumption that they share a similar distribution than the input word's one. They have considered only common nouns for the thesaurus construction that words that appear above 20 times of the frequency threshold were relevant and performed their work based on probabilistic methods using latent semantic indexing (LSI) and latent Dirichlet allocation (LDA). The random projection was selected for dimensionality reduction and they have tested the performance of the thesaurus for query expansion. For 50 number of queries and over more than 170,000 English documents were considered. By selecting the top 10,50,100 thesaurus terms resulted to have the best gain and the precision was above 14% by considering the intrinsic measure. Intrinsic means taking sample words from WordNet and determines how

much it is similarly based on the words co-occurrences. Using the intrinsic measure, a total of 38 neighbors were found from 12243 common nouns using and they used the cosine and mutual information similarity and weighting measures respectively. But using intrinsic measure is not advisable because the neighboring words are limited to a certain synonym set or totally may be absent from the WordNet or Moby reference lists, but extrinsic comparisons are better for accurate query retrieval and expansion. The evaluation of distributional thesaurus through information retrieval tasks has been explored and the performance was tested using different cut-offs using information retrieval and high gain and precision were achieved.

3.3.2 Medical Vocabulary Mining for Japanese

Magnus *et al.* [9] conducted research on medical vocabulary mining using distributional semantics on Japanese patient blogs and random indexing was chosen for extracting terms from the corpus. By segmenting, the corpus into semantic units using a semantic role labeler, and different pre-processing techniques, the authors tried to achieve the goal of the research. Hence Japanese languages are highly agglutinative they first segmented the document to individual lexemes and words into white space so that it is becoming easy for constructing semantic models in which the language requires more window size than Germanic languages. The corpus was collected from TOBYO cite which has three categories medical finding, pharmaceutical drug, and body part. It was contained 270 million characters and 50 million semantic units (2.5 million unique). In addition, with preprocessing they have normalized the syllable writing characters by transforming half-width forms into the corresponding full-width form. Then the corpus was segmented into semantic units by applying the dependency parser CaboCha on the corpus (CaboCha, 2012) and using semantic role labeler ASA (ASA, 2013). There was a total of 12 semantic spaces constructed. Due to its computational efficiency, random indexing was chosen for a distributional model and the cosine similarity measure was used for similarity distance calculation. The performance of the system was not expressed in number but the consistency between recall and precision increases their confidence and the result was promising using distributional semantics.

3.3.3 Medical Vocabulary Expansion for Swedish

Rahman *et al.* [10] conducted research on Swedish medical vocabulary expansion using distributional semantics as a tool to classify the medical terms. The corpus was collected from the journal of the Swedish Medical Association. The authors developed a user interface using a tool that enables them to classify terms from a corpus to a vocabulary list of categories and can facilitate the expansions. The system first checks whether it is a medical term or not yet included in the list of vocabularies using the traditional count method of term frequency-inverse document frequency **(**TF-IDF) and the semantic relation of the terms were done by the distributional model called random indexing with 1000 dimensions of 2 by 2 context window. Using random indexing terms that are far in the semantic distance is rejected. They have evaluated the recall and precision of the new system by comparing it with the previously developed medical vocabularies. By selecting the top 50 candidate terms and compared with other single candidate terms so that the result of the semantic relationship was improved with a recall of 13% than the previous. Therefore, the authors concluded that increasing the top N candidate terms increases the recall and decreases precision.

Skeppstedt *et al.* [44] conducted research on automatic vocabulary expansion of Swedish medical terms using distributional semantics to enhance the development of semi-automatic terminological resources which is very important for medical text processing systems like information extraction and word sense disambiguation. The authors used the Swedish medical text and the subsets of the Swedish version of the medical vocabulary called MeSH. The task was identifying terms that are belonging to a certain semantic category of medical findings and pharmaceutical drugs. Ninety of the terms were correctly classified under the given taxonomic classes. They have implemented a random indexing distributional semantics model for dimensionality reduction as well as for semantic space modeling. The word to word or word to context mapping are assigned a unique identification in the index vector and randomly selected elements are assigned to 0, 1 or -1 in the context vector. Finally, the semantic distances of each word are calculated using the cosine similarity measure. They have extracted the terms using different window sizes like 1+1,2+2,4+4 and 50+50 and it was proved that using large context windows like 50+50 is effective for sentence-level context meanings but it ignores the context definition from adjacent sentences definition. They used term replacement and cosine addition for semantic element extraction by which candidate terms occurring less than 50 times in the document were rejected. Using term replacement with 2+2 window size was showed the best results for medical finding and 1+1 was better for pharmaceutical terms. The context window

size using 2 by 2 dimensions was performing better results for both recall and precision. When retrieving the top n terms, it increases the recall for both the medical finding and pharmaceutical terms in general and results were recorded using the top 1000 retrieved elements with 53% of the 90 expected medical findings and 88% of the 90 expected pharmaceutical drugs.

3.3.4 Distributional Thesaurus Construction for French

Henestroza and Denis [11] conducted research on the automatic construction of distributional thesauri for French-language using FreDist which is a freely available software package for implementation. They have used the freely available L'Est Republican corpus containing 125 million words of journalistic texts. The authors have preprocessed the corpus like tokenization and sentences segmentation using tools, POS tagging using MElt part of speech tagger with more than 97.7% tagging accuracy, lemmatization and morphological analysis using Lefff with words + POS pair to obtain a corresponding lemma and using MaltParser, the fastest and accurate parser.

FreDist has flexibility parameters like - context relation, type extraction, weighing and measuring function, term frequency handling, part-of-speech tagging and filtering of numerical terms. The authors extracted the related contexts as a tuple of three (w, r, w′), where w is the terms in the context and r is the relation of each word and w′ is another word that has a similar meaning with w. The first terms in the tuple may be dependent on the other words in the third tuple list. After this, w is represented as a frequency vector $v^w \in R^d$, where d is the number of unique contexts and v_i^w = freq(w, r,w′), where i corresponds to the context c_i = (r,w′). The term similarity between w_1 and w_2 can be calculated based on the relative frequency of $v^{w1} and\ v^{w2}$ that contains two basic parts called weight and measure function. The former describes the frequency of context relation whereas the second one explains the similarity distance between each candidate terms. The authors used different weight functions like RELFREQ, TTEST, and PMI and measure functions like Cosine similarity, Jaccard similarity, and Line similarity. Even if they used many measures PMI weight function and cosine similarity measure were selected using four parts of speech categories. Once they automatically annotated the corpus with lemmas, POS tags, and syntactic dependency the context relation was extracted and finally unique bigram contexts above the frequency threshold were selected as relevant terms. For evaluation they used two French WordNet as a reference, FREWN which was manually validated and WOLF not manually validated. The first covers verbs and common nouns and the second one contains adjectives and adverbs in addition to the first one. During the evaluation, the author considered only primary lexical terms appearing in both the distributional thesauri

and the WordNet references. It gave them 3,018 common nouns and 1,426 verbs for the FREWN evaluation, and 374 adjectives and 195 adverbs for the WOLF evaluation. During testing the synonyms that are common to the WordNet terms were considered as relevant, and in order to have efficient query retrieval they have used, the combination of bigram and syntactic contexts pairs for better similarity estimates.

3.4 Vocabulary Construction for Ethiopian Languages

Ethiopian languages are languages that the people inside the country speak and communicate. It is known that Ethiopia has more than 80 nation nationalities who speak their mother tongue languages. It includes many local languages like Geez, Amharic, Affan Oromo, Tigrigna, Hadygna, Welaytgna, etc. Many of the people speak Amharic as a second language in addition to their mother tongue and to the reverse there are persons who cannot listen and speak Amharic without their own languages. In some rural areas of the country, people speak their own Mather tongue language. Many of the languages have dictionary lexical resources that are manually built from a corpus, but semantic vocabulary using distributional semantics for enhancing the information retrieval has not been propped so far in any one of the local languages even including Amharic. But there is an attempt done on Amharic thesaurus.

3.4.1 Automatic Thesaurus Construction for Amharic

Andargachew Mekonnen [15] developed automatic thesaurus for information retrieval and query expansion using the Amharic bible. Before the thesaurus terms were extracted, the authors have preprocessed the corpus like stemming, numeric removals and stop words were handled. The author used the stemmer developed by Nega Alemayehu and Willet (2002). In order to generate the thesaurus terms, the word-space model is first searched and the similarity of the terms was done using cosine similarity measure. Latent semantic analysis with SVD was used for semantic modeling and meaning representation. 78% and 22% of the corpus is used for training and testing respectively. The system was tested using information retrieval (IR) for searching and query retrieval. A random selection of terms was showed an encouraging result of 58% accuracy. By integrating the thesaurus with the information retrieval system as query expansion, the recall is increasing with 73.34% and without using it the recall of the system performance is 37.29%.

3.5 Summary

In order to construct Amharic semantic vocabulary, the major areas associated with distributional semantics are briefly reviewed. To accomplish our work the different approaches, techniques, algorithms, models, and APIs are reviewed. Ways how to preprocess and smooth raw texts are assessed. Proper organization and dimension reduction of the raw text corpus are processed for better accuracy, and efficiency. According to the papers reviewed word similarities can be inferred via machine learning, syntactic contexts and term frequency-inverse document frequency (TF-IDF).

Many research works are reviewed in this part. For example, the authors have constructed thesaurus automatically for English but the time complexity was very high and it is not practically scalable for very large corpora. We recommend that using other semantic modeling and dimensionality reduction like Word2Vec may perform better results with relatively less computation time. Some of the researchers worked using common nouns and other parts of speech like adjectives, verbs and adverbs were not considered. It is known that Chinse, Japanese and Arabic languages are highly agglutinative by its nature that the works done for one language cannot be directly applied for other languages including Amharic. The researchers have worked on medical vocabulary expansion for Swedish that the candidate terms were selected by the traditional count method of term frequency-inverse document frequency **(**TF-IDF) which is totally different from meaning extractions using distributional semantics. Once the lexical resource is constructed, it is used for information retrieval by expanding the original query into a set of similar semantic words.

Generally, over the past few years, there have been attempts to use distributional semantics for various natural language processing applications like automatic construction of semantic vocabularies from a given corpus for many languages. Because of the language morphology and its complex nature, research worked for Non-Ethiopian language cannot be applied directly for Amharic. Constructing Amharic semantic vocabulary using a distributional approach and its use for information retrieval is not proposed so far.

Chapter Four: Design of Amharic Information Retrieval

4.1 Introduction

This chapter covers the system architecture that includes components called text preprocessing, word-space modeling, word-space grouping, document indexing and searching with query expansion. The algorithm of some components is briefly described. It also explains about the architecture that defines the system in terms of computational components and interactions among them. The system contains two parts. The semantic vocabulary construction and information retrieval with semantic vocabulary as query expansion. Once the semantic vocabulary is constructed and the documents are indexed it can be accessed offline.

4.2 System Architecture

The system architecture for Amharic semantic vocabulary is composed of text preprocessing, word-space modeling, word-space clustering, document indexing and searching with query expansion. The text preprocessing component takes the Amharic documents and performs preprocessing like tokenization, normalization and stop word removal. Special characters, HTML tags, and non-Amharic alphabets are deleted. After preprocessing, words are tokenized and the spell inconsistency and abbreviated words are normalized into stable units. Stemming is applied to transform the inflectional words into its base words. Then the preprocessed and stemmed words are feed into the word-space modeling which embeds the words across a multidimensional space via the contexts of the words. In order to vectorize the words the parameter like context window size, minimum occurrence count of words, dimension size, workers, architecture, mean value, hierarchical softmax, and negative sampling is considered. Based on these parameters the words are embedded in the vector space. The word-space clustering component calculates the word's angular orientation of cosine distances and clusters the words in certain groups based on their score. It computes the cosine score alongside the words inside the model. Once the semantic vocabulary is constructed it is saved offline to be accessed for information retrieval by expanding the query. The users can search both the semantic vocabulary itself and information retrieval using vocabulary as a query expansion

When the user searches the query, words are compared with the nearest neighbor from the word-space model based on the cosine values, then the top n number of semantically related words are retrieved back to the user. The users are not aware of the expansion of the queries but the system itself reformulates and provides more relevant documents.

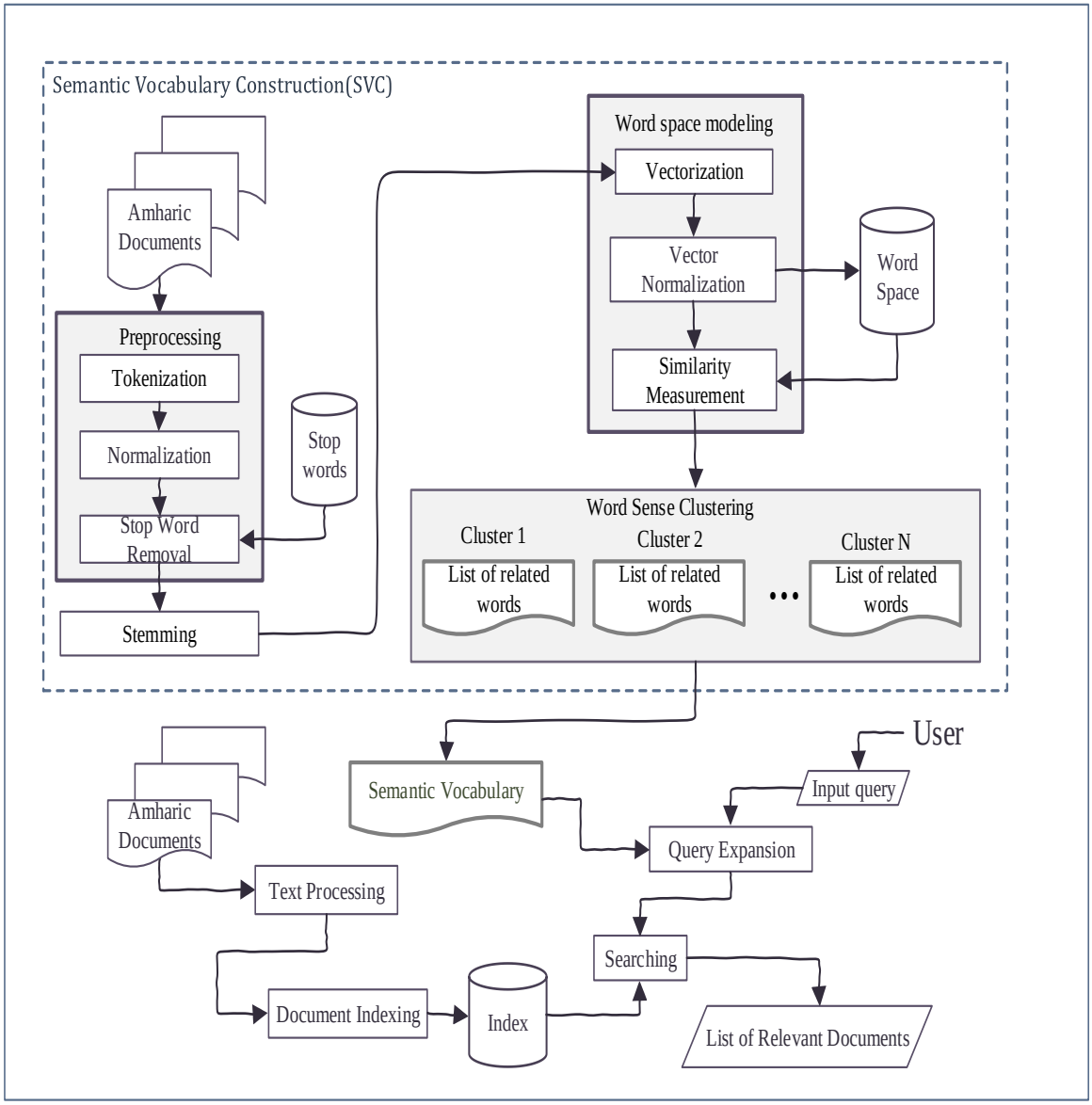

Figure 4.1: System Architecture for Amharic Information Retrieval

4.1 Preprocessing

A document is anything that can be searched or crawled which contains information in different forms like text, image, HTML or XML or plain-texts but, our concern is focusing on text documents known as a corpus. A text document is described in natural language (NLP). The idea here is we take a raw Amharic electronic text documents which are manually crawled from Fana broadcasting corporate, Ethiopian reporter, Z-habesha, Ethiopian broadcasting corporate and Amharic bible.

It is the task of smoothing the raw text documents to make the computation easy and accurate. The reason why we preprocess the data is to increase accuracy, remove noise words and keeping of consistency, to make the data easily understand and interpretable for the models of computations. Normalization and spelling correction are considered especially for alphabetically reach languages like Amharic. It includes the following sub-components.

4.1.1 Tokenization

The individual words are identified as candidates to be used as a feature set for performing a natural language task. A tokenized block of text, usually consisting of indivisible characters are called tokens that do not mean the whole tokenized words of the original text can be used for vocabulary construction. First, the collection of documents in each domain is encoded to UTF-8 texts and then the documents are chopped into individual words. The reason why tokenization is the model is based on the word level embedding and it must build with a set of split words. It is also easy to handle influential words by ignoring the noise words.

4.1.2 Normalization

Alphabets of the same sound or phone but different characters are normalized into one common and probable alphabets. For example, ሀ, ሐ, ኀ, and ሃ are normalized to ሀ, for every word having those alphabets in the corpus. As compared with other languages, Amharic is rich in many varieties of different characters sets but has the same meaning anyone has the chance to write using alternatives as for example ‘a’ and ’A’ has same meaning in English (’አ, ኣ, ዐ, ዓ’), (’ሀ, ሃ, ሐ, ሓ, ኀ, ኃ’),(’ሰ, ሠ’) and (’ፀ, ጸ’) etc. has similar meanings even with different letter inconsistency. Even if the language has its rule of using each character the users are always writing based on their need which makes a challenge for the researchers.

The words are normalized to the same form as ኀይል, ኃይል, ኀይል, ኀይል, ኃይል, ሀይል, ሃይል, ሃይል ኀይል, ሐይል, ሐይል, ሐይል and ኃይል to ሀይል. There are many homophone character in Amharic which includes ሐ, ሑ, ሒ, ሓ, ሔ, ሕ, ሖ,ኀ, ኁ, ኂ, ኃ, ኄ, ኅ, ኆ,ሠ, ሡ, ሢ, ሣ, ሤ, ሥ,ዐ, ዑ, ዒ, ዓ, ዔ, ዕ, ዖ,ጸ, ጹ,

ጲ, ጳ, ጴ, ጵ, ጶ,ፀ, ፁ, ፂ, ፃ, ፄ, ፅ. To avoid the spelling variation problem of Amharic the characters must be mapped into one and common writing alphabets. Not only spell inconsistency but abbreviation like ዓ.ም or ዓ/ም, አ.አ or አ/አ, ዶ.ር or ዶ/ር, መ.ር or መ/ር, ወ.ሮ or ወ/ሮ, አ.ቶ or አ/ቶ, ዳ.ን or ዳ/ን, ት.ቤት or ት/ቤት must be written to their corresponding long forms.

Table 4.1: List of Abbreviations and Its Normalization

List of abbreviations	Normalized to
አ/ም	አመተ ምህርት
አ/አ	አዲስ አበባ
ተ/መ/ድ	የተባበሩት መንግስታት ድርጅት
እ/ኤ/አ	እንደ ኤሮፓውያን አቆጣጠር
ኢ/ዜ/አ	የኢትዮጵያ ዜና አገልግሎት
ኤፍ/ቢ/ሲ	ፋና ብሮድካስቲንግ ኮርፖሬት
ኢ/ያ	ኢትዮጵያ
ኃ/ስላሴ	ሀይለ ስላሴ
ም/አ	ምእተ አመት
ጽ/ቤት	ጽህፈት ቤት
ዶ/ር	ዶክተር
ጠ/ሚ/ር	ጠቅላይ ሚኒስትር
.	.
.	.
.	.
etc	et cetera

4.1.3 Stop Word Removal

Stop words are not considered for vocabulary construction. They are non-content bearing words which have fewer discriminations about a certain document, but useful for the grammar construction of texts (Baeza-Yates and Ribeiro-Neto, 1999). Usually, it has a high frequency occurring in the documents. Stop words like ነዉ, ነበር, ሆኖም, እና, ገለፁ, ዘግበዋል, አስታወቀ, ተናግረዋል, ብለዋል, ከ, ለ, ወደ, etc are common in the document that is not considered as seed terms. None-letter characters like special symbols $, %, &, and extra white space characters, numerals, Non-Amharic characters, and HTML hyperlinks are considered as stop words.

Stop words are extracted automatically from raw text documents based on the number of word frequency up to a certain occurrence and it is manually assessed whether it is a stop word or seed terms. If it is stop word it is stored into a file. Then the content bearing words are filtered out by ignoring the stop lists read from a file.

Input: Amharic documents

Output: list of stop words

```
1.  Begin
2.  Read files from a directory
3.  Determine the top N number of words be filtered out
4.  Make a split of words in the document
5.  Find the words and its frequency up to N
6.  Concatenate the words of each file into one array
7.  Display the words with its frequency
8.  Manually assess whether it is the stop word or not
9.  if stop word:
        jump
10. else:
        delete the words
11. Store the list of stop words into the disk
12. End
```

Algorithm 4.1: Filtering Stop Words

Input: Amharic text document

Output: Purified Documents

```
1. Begin
2. Read the documents from a directory
3. If read:
       3.1   Perform tokenization
       3.2   Perform normalization
4. Read stop words from a disk
5. Goto to step 3.1
6. Goto to step 3.2
7. If word found in normalized words at step 3.2:
       7.1   Remove the word
8. Else:
       8.1   End of file
9. Write the file into the disk
10.    End
```

Algorithm 4.2: Stop Word Removal

4.2 Stemming

The inflectional words must be transformed into the base words which are the usual and prior tasks for natural language processing for morphologically rich languages like Amharic. A stemmer is developed by Nega and Willet [45] that removes the inflectional and derivational affixes. For example, the document "የእጅ ጣታችንን ስልክ የሚያስጠቅመው ቴክኖለጂ ኤስጂኤንኤል ሰኣት በእጃችን የሚታሰር ቴክኖሎጂ" stemmed to "ጣታችንን ስልክ ያስጠቅመው ቴክኖለጂ ኤስጂኤንኤል ሰኣት እጅ ቴክኖሎጂ". Hince the stemmer is no much efficient some of the incorrectly stemmed words must be transformed into its associate forms. For example, ውጤቶ is transformed into ውጤት. Words like ስልኮ, ገፆ, ፊረንጆ, ኮምፒውተሮ, ደንበኞ, መስመሮ, ባንኮ, ሶፍትዌሮ, ቁጥሮ, አማራጮ, ታብሌቶ, ካርዶ, ጥቃቶ, ፋይሎ must be transformed into its normal words by changing the last letter into the six-order character of its corresponding '*sads'* word.

4.3 Word-space Modeling

This component contains computational tasks called vectorization, vector normalization, and similarity measure. After the Amharic text document is preprocessed and stemmed the words are extracted across a row, and documents across a column of the vector space with a reduced dimension of the contextual window. The minimum frequency occurrence count is set to an integer number to consider the words in the model. The dimension of the word-space model is determined by the number of words to be plotted across the N-dimensional space. To get the most important features of the document dimension reduction is performed. Even though the largest dimension size has fewer performances for large document sizes, but for a better vector representation, the dimension size must be between [200,300].

Center Word and Context Word Identification

CBOW extracts the target or center words for given context words based on the context windows sizes. The meaning of words is extracted based on the window size. Increasing window size gives more context words. The Word2Vec model has the arguments of the input text, sample size, context windows size, and occurrence count that enables for semantic distance calculations. There are different notions of contexts like windows around the word and dependency-based features Tim Van de Cruys (2015). The first is based on the number of words surrounding in related meaning and the second is based on the syntactic patterns of the words with similar meaning with similar cooccurrences. After the content bearing words are identified it is learned with the model based on its contexts with the assumption semantics can be extracted by words surrounding the words. Finally based on semantic distance calculation

top most similar number of words across each word are clustered into a set of sense groups. The word space modeling considers the following three processes.

4.3.1 Vectorization

It represents word contextual occurrences in the document in the form of numerical values across N-dimensional spaces. It is done with a continuous bag of words (CBOW) architecture, due to its applicability for small document size. To vectorize the words across N-dimensional space the following parameters are considered.

- ✓ **Frequency Weighting**

It is the measure of how often and how many times a term is occurring in a certain text corpus. The computing of term frequency is identifying how many times of each word happen in the document collection. This concept touches the minimum occurrences of terms in a document to be considered for vector representations.

- ✓ **Dimensionality Reduction**

It improves the utility in memory-constrained devices, especially when running on the user's personal computer which has limited space and computation power. To reduce the size of embedding it is possible to use the Word2Vec dimensionality reduction up to a certain size. It saves memory space but affects the analog information. For example, to save memory for 2.3M data using a 300-dimension vector on a 64-bit system it consumes over 6 GB memory.

- ✓ **Context Window Size**

It determines how many words before and after a given word would be included as context words of the given word. For words in the row, the matrix is mapped within the context window of the column matrix to calculate the maximum value of word co-occurrence contexts. The size of the context windows can be one window, two windows like 2+2, several windows e.g. 3+3,4+4 or the range of several context windows like 1-10. We can increase or decrease the sizes to get a better match for the target or context word extraction from a set of documents. For example, assume the source text document is "**የኢትዮጵያ ኦሊምፒክ ኮሚቴ 43ኛ መደበኛ ዓመታዊ ጠቅላላ ጉባዔውን ያስተናገደበት መድረክ መቼ ነው የተዘጋጀው?"** after performing the text operation the source text becomes **"ኢትዮጵያ ኮሚቴ ጠቅላላ ጉባኤ መድረክ መቼ"**. Then the Word2Vec model learns the context words like in the following table with windows size two.

Table 4.2: Example of Context Word Vectorization

ኢትዮጵያ	**ኮሚቴ**	**ጠቅላላ**	**ጉባኤ**	**መድረክ**	**መቼ**	**Implies**	**Training when window size=2**
ኢትዮጵያ	ኮሚቴ	ጠቅላላ	ጉባኤ	መድረክ	መቼ	➡	(ኢትዮጵያ, ኮሚቴ) (ኢትዮጵያ,ጠቅላላ)
ኢትዮጵያ	**ኮሚቴ**	ጠቅላላ	ጉባኤ	መድረክ	መቼ	➡	(ኮሚቴ, ኢትዮጵያ) (ኮሚቴ, ጠቅላላ) (ኮሚቴ, ጉባኤ)
ኢትዮጵያ	ኮሚቴ	**ጠቅላላ**	ጉባኤ	መድረክ	መቼ	➡	(ጠቅላላ, ኢትዮጵያ) (ጠቅላላ,ኮሚቴ) (ጠቅላላ, ጉባኤ) (ጠቅላላ,መድረክ)
ኢትዮጵያ	ኮሚቴ	ጠቅላላ	**ጉባኤ**	መድረክ	መቼ	➡	(ጉባኤ, ኮሚቴ) (ጉባኤ,ጠቅላላ) (ጉባኤ,መድረክ) (ጉባኤ,መቼ)
ኢትዮጵያ	ኮሚቴ	ጠቅላላ	ጉባኤ	**መድረክ**	መቼ	➡	(መድረክ, ጠቅላላ) (መድረክ,ጉባኤ) (መድረክ,መቼ)
ኢትዮጵያ	ኮሚቴ	ጠቅላላ	ጉባኤ	መድረክ	**መቼ**	➡	(መቼ, ጉባኤ) (መቼ,መድረክ)

All the documents from different domains learn as in the same way this table shows above.

Word2Vec embedding parameters are not limited to frequency weighting, dimensionality reduction, and context window size but also considers the following.

- ✓ **Training Architecture:** the continuous bag of words (CBOW) architecture is fast and basically used for frequent words than skip-gram which is slow and better for infrequent words. Hence Amharic is morphologically rich language and the words can be transformed into many stemmed words from a single root word it applies the CBOW architecture.
- ✓ **Training Algorithm:** applying the negative sampling enables to remove noise words. It is better for frequent words and low dimensional vectors than hierarchical soft-max which is better for infrequent words.
- ✓ **Sample:** the higher-frequency words are normalized into a set of average integer numbers.

- ✓ **Workers:** this is how to use multiple threads to fasten the training algorithm with parallel processing. It is also possible to use worker value equivalent to the core of the system.
- ✓ **Iteration:** is the number of epochs over the corpus to speed up the training. Instead of learning the entire document at once, dividing for a certain iteration of integer numbers can fasten the training.
- ✓ **Sorted Vocabulary:** semantic vocabulary is a lexical resource that users can search similar words. To save their time of searching and looking of relevant results the vocabulary must be sorted in the descending order.

4.3.2 Vector Normalization

It implies computing the semantic distances between vectors by forgetting the previously learned vectors and it is performed after the training is finished. If the vector is not normalized it will be trained iteratively without ignoring the original vectors in addition to the recent words. So, we must apply normalization for better performances and memory use.

4.3.3 Similarity Measurement

After the words based on their context is learned and normalized, a semantic similarity measure is applied. It is based on the cosine values which are implemented with vector orientation, not magnitude. The higher the score the higher word relatedness will be built. Values approaching 1 is highly related and values when it is 0 it is totally opposite. Based on their orientation words close to each other are grouped into a certain semantic cluster.

4.4 Word Sense Clustering

This component works automatically clustering of vectors of similar terms across the n-dimensional spaces. It is performed after the words are mapped on the vector spaces and the semantic closeness is calculated. In this sense, it is defined as the clustering of semantically related words based on cosine values into a set of sense groups without predefined labels or categories. It is performed after vectorization and normalization are done.

```
Input: The learned word-space vector files
Output: list of similar words
1. Begin
2. Import the learned vector space
3. Determine the topn semantic word-groups
4. Calculate the semantic distance values
5. Write to the disk
6. Commit writing
7. End
```

Algorithm 4.3: Semantic Vocabulary Construction

4.5 Information Retrieval

It is looking for relevant items by providing a query from a collection of lexical resources like semantic vocabulary and form indexed data sources. This section basically talks about semantic vocabulary searching and information retrieval from indexed data files expanding the query terms to enhance retrieval relevancy.

4.5.1 Vocabulary Searching

Once the semantic vocabulary is constructed from different domains the words can be searched either in the help of graphical user or command-line interface. But using a graphical user interface makes the system easy and interactable for the users. Users have also the option to select the top n number of similar words and the domains. First, the user provides the query words then it is applied text operation. Then words are searched from the semantic vocabulary based on their nearest neighbor of the terms using cosine distances.

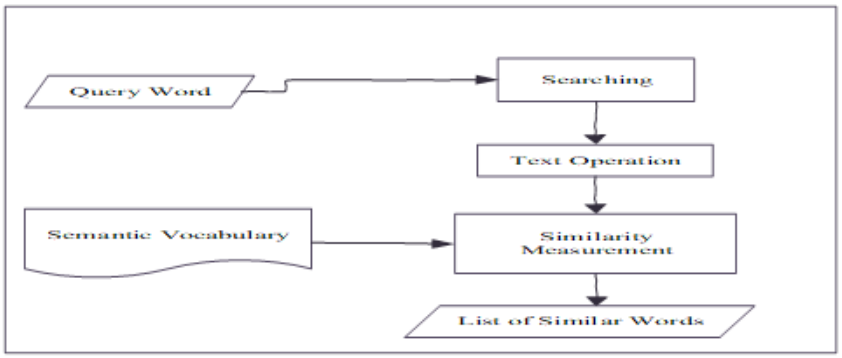

Figure 4.2: Architecture for Vocabulary Searching

```
Input: Query word
Output: List of similar words
1. Begin
2. Enter the query word
3  Determine the topn similar words to be returned
4  Select the domain #Sport, Technology, Religious, General
5  Applying text processing
     If words exist in the semantic vocabulary:
          Calculate similar words based on cosine measure
          Return similar words
     Else:
          The query word is not in the vocabulary
6  End
```

Algorithm 4.4: Semantic Vocabulary Searching

4.5.2 Information Retrieval with Semantic Vocabulary

Information retrieval is the intent that the user seeks relevant information for his or her query terms by expanding from the semantic vocabulary to retrieve matched results from indexed sources. First, the query words are split into unique tokens and it expands the words into its corresponding contextual similar words from the semantic vocabulary. Then the score of the query in the documents is calculated based on its frequency, TF-IDF, and BM25F, then most relevant documents could be retrieved according to its rank in the descending order. Expanding the information retrieval system with semantic vocabulary increases the number of relevant documents to be retrieved.

Text Processing

Firstly, the Amharic text documents are read from the directory and applied text operations like tokenization, normalization, stop word removals and stemming.

4.5.3 Document Indexing

It is the process of associating the information with a file to make the retrieval system efficient. Search with indexed data saves time and easy to handle search results. The values in the index are associated with the filed elements. Before the documents are indexed the metadata or the field elements must be considered. Because the filed elements make the searching easy. For

example, searching the name of the book via google is better to consider the ISBN-number or the title and topic rather than a silly reading of the entire contents in the book. So the documents can be indexed by considering the field elements like path, title, content, and date or time. Then after the documents are indexed it is accessible offline for later retrieval.

```
Input: Amharic text document collections
Output: Indexed file
1. Begin
2. For file in reading text document from a directory
   2.1  If successfully read:
        2.1.1   Apply text preprocessing
        2.1.2   Specify the field elements
        2.1.3   Associate the field elements with each file in
                a directory
        2.1.4   Write into the indexer
   2.2  Else: End of file
3. Commit writing
4. End
```

Algorithm 4.5: Document Indexing

4.5.4 Query expansion

The original query is reformulated to retrieve more relevant documents. It adds more query terms from the semantic vocabulary. Queries can be expanded from WordNet, thesaurus, database, ontology and any lexical resources either automatically created or manually built. But manually built resources takes time and are laborious. The nature of the resource and queries also matters to the retrieval system when it is expanded. Phrases or sentence level queries can retrieve better than a single word. Because at the phrase level when the words are split and expanded the number of query words becomes increasing. Semantic vocabulary is created based on the assumption that words surrounding anther words tend to have similar words. This is essential when the size of the data is huge and meanings can be extracted contextually. After the documents are indexed the user provides the query terms by expanding automatically from the semantic vocabulary. Then, the expanded query terms are searched from the index and relevant documents are retrieved for the user based on the score for query terms and document ranks.

4.5.5 Document Searching

Once documents are indexed it is searched. First, the users provide the query then the system applies test processing. When using ranking based information retrieval, the relevance of the query terms in the documents is calculated to retrieve the top relevant documents in the descending order of its ranks. The scores are based on Boolean search approaches. It includes OrGroup, AndGroup and Not. Using OrGroup best results can be retrieved. Unless all the query terms do not much it can return a result. If the query terms match in any one of the field elements the relevant documents will be retrieved. Using AndGroup it searches the documents exactly containing the query terms and results will be less relevant and retrieved. Using Not Boolean searching documents containing certain query but not any other will be retrieved. After matching for query terms is found the rank of relevant documents, the path of the file, the term scoring, the title, and the text data is retrieved for the users.

```
Input: Amharic text queries
Output: List of relevant documents
1. Begin
2. User enters query
3. Split the query terms
4. Applying text processing
5. Check whether the query term is much to the semantic
   vocabulary?
  5.1 if doesn't match:
          continue to next query terms
  5.2 else:
        5.2.1  Expand the queries from semantic vocabulary
        5.2.2  Search from indexed data
        5.2.3  Calculate the query score in the documents
        5.2.4  Rank the documents based on their score
        5.2.5  Display relevant documents in descending
               order
6. End
```

Algorithm 4.6: Document Searching

Chapter Five: Experiment

5.1 Introduction

This chapter discusses the system implementation, document collection, prototype and evaluation of the study. Automatic construction of semantic vocabulary is implemented in nine domains containing documents collected manually from Ethiopian public media like Fana broadcasting corporate, Ethiopian reporter, Ethiopian Broadcasting corporate, Z-habesha.com, Ethiopian telecommunication, and Amharic bible. Many of them are article documents other than the religious domain.

Since stemming plays a vital role in vocabulary construction, we have applied these text operations done by other researchers. The stop words are filtered out automatically from the documents for each domain and manually assessed. Hence the stemmer is not much efficient we filtered out the incorrectly stemmed words from each domain and it is transformed into the correct form.

It also explains the visualization of similar and dissimilar words across a vector. Because visualizing using figures is more appropriate and easier to understand. We have done an evaluation of the system in terms of intrinsic and in the area of information retrieval which can enhance the number of relevant documents to be retrieved.

5.2 Corpus Collection

It is undoubtedly known that the web is the most important source of texts for different languages as it is for Amharic too, and the documents were collected manually from unstructured Amharic news articles of different domains from the Ethiopian public media like Fana broadcasting corporate, Ethiopian reporter, Amharic holy bible, and Ethiopian telecommunication. 8,759 Amharic documents are collected as shown in the following table.

Table 5.1: Collected Amharic Documents from Different Domains

Domains \ Number of documents		Total documents
Religious**:** Amharic Bible	Old Testament (929) New Testament (260)	1,189
Fanabc.com	General Domain (519)	519
Ethiopian Reporter	Business (1,317) Sport (700) Politics (1,002) Law (1,037) Art (1,016) Health (272)	5,344
Ethio Telecom	Technology (978)	978
Z-habesha	Health (529) Sport (200)	729
Total Documents		8,759

5.3 Implementation

The system is implemented using python programing language it is fast and efficient for data-oriented processing. We have applied the Word2Vec semantic API of the gensim library, the Spyder editor and the Qt-designer for a graphical user interface (GUI).

Stop Words Filtration

The stop words are filtered out on the assumption that high frequently happened words are insignificant to the semantic understanding of documents rather than the grammar formation of the syntax. There are no standard and ready-made stop words for the Amharic language. 1200 stop words are extracted from nine different domains as shown in the following samples.

ገልጸዋል። ፡1264	እንዲሁም፡ 727	ናቸው።። 637	ይችላል፤»፡ 34
ብለዋል። ፡1262	ብቻ፡ 721	በተለይ፡ 604	ይገኝበታል።። 30
ደግሞ፡ 1200	እንደ፡ 710	ታውቋል።። 377	አሳስቧል።። 29
ነው፡ 1204	ጠቁመዋል። 648	ይባላል።። 68	ይዟል።። 23

Figure 5.1: Sample Stop words

Amharic Stemmer

As it is discussed so far, we have implemented the Amharic stemmer developed by Nega and Willet *[45]* that some of the incorrectly stemmed words are transformed into its proper forms. Some of the words containing the last letter in the six orders of the Amharic alphabet is changed to seventh order. For example, መስመሮ, ባንኮ, ሶፍትዌሮ, ቁጥሮ, አማራጮ, ታብሌቶ, ካርዶ from the Ethiopian telecommunication domain is transformed into መስመር, ባንክ, ሶፍትዌር, ቁጥር, አማራጭ, ታብሌት, ካርድ which is shown in the following table.

Table 5.2: Incorrectly Stemmed Words

Domains	**Number of incorrectly stemmed words**
Technology	70
Religion	40
Politics	87
Law	30
Sport	27
Art	200
Business	61
Health	43
General Domain	20
Total	578

5.4 Word Embedding Parameters

Word2Vec implements parameters that affect both training speed and quality of word representations. Among these, we have considered the following parameters.

- ✓ ***Training Architecture***: we have implemented the continuous bag of words (CBOW) architecture because of its fast and basically used for frequent words than skip-gram which is slow and better for infrequent words.
- ✓ ***Training Algorithm:*** we have implemented negative sampling. Because it is better for frequent words and low dimensional vectors than hierarchical soft-max which is better for infrequent words.
- ✓ ***Vector Size:*** we have implemented with 300-dimension size, but possible to use up to 400 and it depends on the size of the corpus. Usually, more is better but not always.
- ✓ ***Window Size:*** It is the maximum distance between the target word and the context word and we have implemented the size into 10. The default values for CBOW=5.
- ✓ ***Minimum Count:*** it is the frequency of words that how many times it appears in the corpus and words above 5 integer values were considered.
- ✓ **Sample**: to downsample higher-frequency words we set the threshold 1e-4 which can be possible values 0 up to 1e-5 and default 1e-3
- ✓ ***Workers:*** this is how to use multiple threads to fasten the training algorithm with parallel processing. It is advisable to use the worker value equivalent with the core of the system and in our case, we have applied the `multiprocessing.cpu_count()` which automatically runs threads parallel with the number of the system CPU.
- ✓ ***Sorted Vocabulary:*** we have sorted the vocabulary, so the most frequent words have the lowest indexes.
- ✓ ***Iteration:*** is the number of iterations or epochs over the corpus to speed up the training. We set the values to 20 but Word2Vec is not limited to the above parameters but also there are others like mean averaging for enabling better vector representations.
- ✓ ***Negative Sampling:*** we *have* used the integer values of negative sampling to be 10 for specifying how many noise words should be ignored.

Table 5.3: Learning Time Taken

Domain	Learning Time
Politics	10 seconds
Business	14 seconds
Sport	08 seconds
Art	15 seconds
Law	11 seconds
Technology	12 seconds
Health	09 seconds
Religious	12 seconds
General Domain	07 seconds

Vector Representation of Word

The way in which words with its context is represented using real numbers across a multi-dimensional space and the meaning of similarity of words is calculated from their appearance. For example the word "ኢትዮጵያ" is represented across a vector space via 300 dimension contextual vectors like: ኢትዮጵያ [0.22846702 -0.18636888 0.37463588 -0.39620417 0.06327345 0.27090612 -0.02564 -0.08138988 -0.31631055 -0.33241597 -0.08660156 0.22223112 -0.17032604 0.08228066 -0.14271669 0.22033538 -0.17548716 0.023884868 0.07955296 0.22194546 0.043573804 -0.05689095 -0.043901976 -0.030364705 -0.14126454 -0.08972345 -0.09260801 -0.08430293 0.38275287 -0.072490625 0.09843942 0.07864776 0.26113772 0.047812853 -0.2324682 -0.13765712 0.13860032 0.0564319 0.36669126 -0.4215609 0.11712133 -0.1097 -0.08198221 0.005150028 -0.6395629 0.44110343 -0.09969088 0.19976714 0.21972564 0.21426888 0.13665165 -0.18646276 -0.5661503 0.08119833 -0.23518856 0.5245922 -0.006311839 0.11442948 0.33685833 -0.4472682 0.14937675 0.019622786 0.08172888 -0.4614889 -0.0331291 0.24178408 -0.44752008 -0.3320526 -0.17595649 0.19702989 0.24322465 -0.44135997 -0.059035245 -0.3639248 -0.016715415 0.11519881 -0.061159894 0.12039552 -0.27031842 0.21330291 0.29706654 -0.48732874 0.08427961 -0.109723516 0.17713596 -0.123053715 -0.15194054 -0.35676602 0.09761115 -0.45297745 -0.1700975 0.034428008 -0.1282237 0.4602612 0.16188802 0.08635316 0.19495748 0.40158692 -0.20867911 -0.28360382 0.034472506 -0.01613345 -0.089673825 0.16306269 0.28832775 0.006294617 -0.58605134 -0.24795212 -0.3071882 -0.28161737 0.31483188 -0.33917677 -0.033043537 0.019884286 -0.09990305 0.09698222 -0.56061876 0.18900803 -0.1873427 0.06999493 -0.5538926 0.17531055 -0.10930416 -0.19572714 -0.16587967 -0.2134034 0.12141156 0.53872615 0.15840912 -0.08756662 0.08972515 -0.22902273 -0.3773744 -0.2546918 0.38887897 0.59770423 0.17608327 -0.28503403 0.33630255 0.05659163 0.27847502 0.12224267 -0.0102872765 -0.4059241 -0.25153035 -0.19720684 -0.32420614 0.14603814 0.098454416 -0.34335777 0.010991277 0.21577989 -0.54554373 -0.37783274 0.30433914 0.028718902 -0.21841204 0.001716722 -0.026639825 0.2039842 -0.033317316 -0.04912324 -0.12318634 -0.3067145 0.28923675 -

0.08171267 0.2288205 0.17652266 0.29787025 0.41852096 -0.2430533 0.04617029 -0.07332684 -0.29282156 -0.1310922 0.105451316 -0.69727415 0.17842904 -0.69105357 0.32440943 -0.30494258 0.02200511 0.19290456 0.18123962 -0.21798196 0.52945226 -0.52420014 -0.1801077 -0.29652873 0.1194064 -0.23854184 -0.006795387 0.27029416 -0.010333688 -0.00013160828 0.24778005 0.0069759665 -0.28492466 0.45527098 0.030055199 0.22923505 0.30260134 -0.35357565 0.050794534 -0.14784622 0.13801818 -0.03827206 0.14471923 -0.3578073 -0.08733592 0.038370155 -0.14262515 -0.14457095 0.4241994 0.3510078 -0.4278105 0.5275858 0.48712322 -0.10989516 0.47417527 -0.1704997 -0.08501538 -0.11221606 -0.25410908 0.44532162 -0.27760726 -0.19585255 0.29126456 -0.15276703 -0.19725555 0.029256536 -0.24701335 0.25828868 -0.2257213 -0.082018726 -0.26193357 0.28266144 -0.09618708 -0.09462489 0.14584225 -0.2180487 -0.038949974 0.069947176 -0.4171325 -0.058855712 0.16912185 0.21025158 -0.39612174 -0.29776782 -0.48376587 -0.3887403 -0.038175043 -0.2769305 0.018514274 0.021224244 0.46774945 0.4967832 -0.06437573 0.06749164 0.064404614 0.4440048 0.009197 0.012561805 -0.17443249 -0.3103287 0.026123893 -0.094862826 -0.22791566 0.23518895 -0.54600954 0.5051063 -0.18003993 -0.32822096 0.06843247 -0.65230954 0.015825689 0.04142233 0.109396845 0.047601365 -0.4369641 -0.016632654 -0.16340087 0.3530912 -0.26066202 -0.4191187 0.3576366 -0.100643255 -0.34647936 -0.08448479 -0.08563565 0.08858384 0.30538854 0.39992473 -0.16235532 -0.060552225 0.10500662 0.23440412 0.29167837 -0.25117168 0.18234825] and to make simply visible it is represented words by context like the following.

ኢትዮጵያ [0.22846702 -0.18636888 0.37463588 -0.39620417 0.06327345 0.27090612.....]

ውድድር [0.28687748 0.096131936 -0.22931051 -0.1902187 0.18935864 0.11344301…….....]

ኳስ [0.122336544 -0.01815366 0.28075188 -0.17352425 -0.054913472 0. 028958626……....]

ቡድን [0.0027367843 0.40312245 0.026420604 -0.025145505 0.08414483 -0.0131022……....]

ፌዴሬሽን [0.17593175 -0.47367355 0.16456302 -0.32902747 -0.2203794 0.047293644…. …..]

ኦሊምፒክ [0.17748585 -0.3580443 -0.15674385 -0.30170944 0.05036655 0.08042107.....]

Word Vector Visualization on a Two-dimensional Plane

Visualization is essential to make big data easier for the human brain to understand, view in naked eyes, detect patterns and trends. It can also clearly inspect how big data is vectorized using machine learning approaches. We have implemented the distributed stochastic neighbor embedding(t-SNE) algorithm. Because of t-SNE can handle nonlinear data efficiently than others like principal component analysis (PCA). The vector representation for 200 sample words from the model across a vector space is shown in the following figure.

Figure 5.2: Word Vector Visualization using t-SNE

5.5 Vocabulary Construction

A semantic vocabulary is constructed after the parameters of Word2Vec is clearly defined and the words are vectorized across a vector space using 300 dimensions of different domains. Then based on the word's orientation or appearances the semantic clustering is performed. This is done by calculating the angular occurrences of words by applying the cosine implementation by which the model has the most_similar() and most_similar_cosmul() methods that are better than cosine, but still implements cosine metrics and words with similar or nearly similar contexts are became under a certain group. For example, the semantic distance value for 'ስልክ'is closer to the words 'አመታት', 'መተግበሪያ', 'ነፃ', 'ሞባይል'. After the vocabulary is constructed users can search by providing the queries and determining the top n similar words. Then the system responds the semantic similar words for the user's query. The full searching and accessing of the semantic vocabulary are clearly depicted in figure 5.7 and the following figure shows sample similar words for the query words with the top fifteen number of words to be searched.

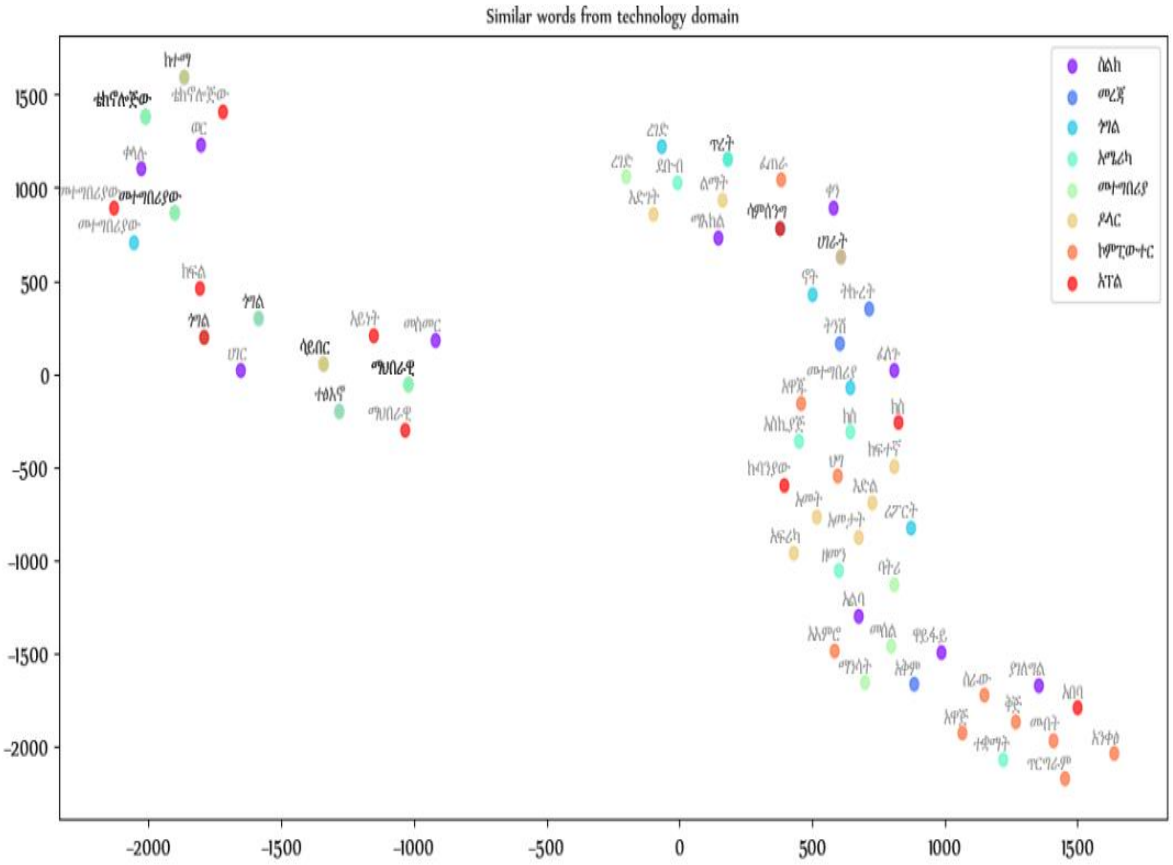

Figure 5.3: Groups of Semantically Related Words

{'**ስልክ**': ['አመታት', 'መተግበሪያው', 'ቡክ', 'ነፃ', 'ደንበ', 'ሞባይል', 'ጥናት', 'ኔትወርክ', 'መጠቀም', 'ትንሽ', 'ከፍተኛ', 'መተግበሪያ', 'ቀን', 'ደንበኛ', 'ምክር'], '**መረጃ**': ['ተሳፋሪ', 'አመታት', 'ቡክ', 'ክፍል', 'ሰው', 'ልጅ', 'ቁጥር', 'ኢንተርኔት', 'ጥናት', 'አንድ', 'ዘመናዊ', 'እድሜ', 'አይነት', 'ሁለተ', 'ሀሳብ'], '**ንግል**': ['መተግበሪያው', 'ሀሳብ', 'ነጥብ', 'ጊዜያት', 'ክፍል', 'ቡክ', 'ከተማ', 'መጠቀም', 'ኢንተርኔት', 'ተጠቃሚ', 'ባለሙያ', 'ዘዴ', 'ጥናት', 'አይነት', 'ምክር'], '**አሜሪካ**': ['ጥናት', 'አመታት', 'ሀገሪቱ', 'መልእክት', 'ቴክኖሎጅው', 'ቁጥር', 'ዘመናዊ', 'ቀን', 'አይነት', 'አማካይነት', 'ቡክ', 'ቤቱ', 'ማህበራዊ', 'ሰው', 'ባለሙያ'], '**መተግበሪያ**': ['መጠቀም', 'ባለሙያ', 'ጥናት', 'ቀን', 'ክፍል', 'ሰው', 'ቁጥር', 'መተግበሪያው', 'አመታት', 'ከፍተኛ', 'ቡክ', 'እድሜ', 'ልጅ', 'ቅስቃሴ', 'መልእክት'], '**ዶላር**': ['መጠቀም', 'አመታት', 'ከተማ', 'ስልክ', 'ኔትወርክ', 'ሞባይል', 'ጥናት', 'ትንሽ', 'ወያላው', 'መተግበሪያው', 'ውጭ', 'ነጥብ', 'አይነት', 'ቀን', 'ትልቅ'], '**ኮምፒውተር**': ['ፕርግራም', 'አዋጅ', 'አንቀፅ', 'ህግ', 'ቅጅ', 'አዋጁ', 'መብት', 'አንድ', 'ኮምፒውተሩ', 'ውጤት', 'ነጥብ', 'ፈጠራ', 'አመታት', 'ማሳየት', 'አይነት'], '**አፕል**': ['ባለሙያ', 'አንዱ', 'አመታት', 'ትልቅ', 'ደንብ', 'ሰል', 'ስልክ', 'መጠቀም', 'መተግበሪያ', 'ታሪክ', 'ቁሳቁስ', 'ጥናት', 'ግንኙነት', 'መተግበሪያው', 'ቴክኖሎጅ']}

5.6 Semantic Clustering

The system returns the corresponding semantically related words based on the cosine values after specifying the top N number of similar words for the input query terms. Clustering in this concept defines grouping of semantically related words into a set of one semantic fold without specifying the label or category of the data. So, we have clustered the semantic vocabulary using gensim distances. As it is discussed in chapter two, there are a few semantic distance calculations but the gensim semantic distance is the one which is a cosine implementation of Word2Vec that has better results than Jaccard and Euclidian as it is discussed in table 5.4. The following picture shows sample semantically related groups of words for topn= 12

{'**ማንቸስተር**': ['ግጥሚያ', 'ዩናይትድ', 'ሲቲ', 'አርሰናል', 'ቼልሲ', 'ሊቨርፑል', 'ማንችስተር', 'ሻምፒዮንነት', 'አማካይ', 'ሲዝ', 'አጥቂ', 'ሪያል'], '**ቻምፒዮንስ**': ['በማስቆጠር', 'ሮናልዶ', 'መረብ', 'ክርስቲያኖ', 'ቶተንሀ', 'ችሏል', 'ሌስተር', 'ኤቨርተን', 'እንግሊዝ', 'ለማስቆጠር', 'ዘንድሮ', 'አውሮፓ'], '**አርሰናል**': ['ቼልሲ', 'አማካይ', 'ማንቸስተር', 'ሲዝ', 'ሊቨርፑል', 'መስመር', 'ግጥሚያ', 'ማንችስተር', 'ተከላካይ', 'አጥቂ', 'ዩናይትድ', 'ሲዘን'], '**ቸልሲ**': ['ኒውካስትል', 'ዳኒ', 'ማንቼስተር', 'ደሞ', 'ሆላንዳዊ', 'ሉካክ', 'አቻነት', 'ሲልቫ', 'ፎርሜሽ', 'ሳንቼዝ', 'ሮማ', 'ሴስክ'], '**ፈይሬሽን**': ['አስመራጭ', 'አስመልክቶ', 'ሰብሳቢ', 'ዘሪሀ', 'አባላት', 'ስራ', 'ሪፖርተር', 'ይግባኝ', 'ዝርዝር', 'መተዳደሪያ', 'ውክልና', 'ፊፋ'], '**ኮሚሽነር**': ['ጸሀፊ', 'አላዩ', 'መልእክት', 'ሚናገሩት', 'ቀደም', 'ያቀረበው', 'አስኪያጅ', 'አጀንዳ', 'ድምጽ', 'ቅጥር', 'መደረጉ', 'ሰሚ'], '**ሽያጭ**': ['እያለ', 'በይበልጥ', 'የተነሳም', 'ጋዜጣ', 'ተፈጥሮዋዊ', 'የገለጸ', 'አያንስም', 'መሳሪያ', 'ጥቃት', 'የሚረዱ', 'በመጣራት', 'መሸጥ'], '**ኡመድ**': ['ማለዳ', 'ተሸመ', 'ስጢፋኖስ', 'የሚደረጉት', 'አግኝተዋል', 'ሰኢድ', 'ላንጋሮ', 'ቆይቶ', 'ጌታነህ', 'ናስር', 'አራዳ', 'ሽመልስ']} for terms in ['**ማንቸስተር**', '**ቻምፒዮንስ**', '**አርሰናል**', '**ቸልሲ**', '**ፈይሬሽን**', '**ኮሚሽነር**', '**ሽያጭ**','**ኡመድ**']}

Figure 5.4: Semantically Related Groups of Word

Table 5.4: Semantic Distance Values Between Words

Metrics / Words		Cosine	Euclidian	Jaccard	similarity()	most_similar()	most_similar_cosmul()
ሲቲ	ዩናይትድ	0.9966782888565162	0. 9184928042850455	0.0	0. 9966782888565155	0.9966784119606018	0. 9983382225036621
ተጨዋች	ዝውውር	0.9582914049854455	0.7111796614828165	0.0	0.9582914049854449	0.9582913517951965	0.9791446924209595
ወርቅ	ሜዳሊያ	0.9993390272711175	0.9636414298402286	0.0	9993390272711162	0.9993390440940857	0.9996686577796936
አፍሪካ	ሱዳን	0.9649903627026328	0. 7353884587325511	0.0	0.9649903627026328	0.9649903774261475	0. 9824941754341125
ዳኛ	ፍትህ	0.9032567669142901	0.5601290314661848	0.0	0.9032567669142895	0.9032567143440247	0.9516274333000183

The Euclidian distance is based on the magnitude of vectors but the cosine distance is based on the angle between vectors. The angle measure of vectors is more resilient to the frequency of term counts which are semantically related and the magnitude measure is affected by frequency occurrences. Measuring semantics of high dimensional data using Euclidean distance is useless relative to cosine similarity. Jaccard distance is not advisable to use in vector comparisons. But there are gensim vector distance calculations like most_similar and most_similar_cosmul methods which are provided by the Word2Vec implemented with python by which the second one is better than the rest of others as the values are specified in the table above.

The cosine, Euclidian and Jaccard distances are an extension of scipy.spatial.distance but most_similar() and most_similar_cosmul() are the gensim Word2Vec implementation. Word2Vec also implements directly the cosine metric using similarity() methods when we want to compare a set of words. Hence the most_similar_cosmul() similarity method is better and some of the Word2Vec distance is an implementation of cosine values which works on vectors, not magnitude. We have considered and implemented the gensim distances for vocabulary construction.

The distance measure does not affect the number of semantically related words, but the higher the values the more it is semantically close or the degree of closeness becomes high. For example for the term ‘ዲባባ’ the top 5 number of semantically related words when using most_similar() are: [('ጥሩነሽ', 0.998961329460144), ('ቀነኒሳ', 0.9960207343101501), ('ሜትር', 0.9956890940666199),('አልማዝ',0.9944643974304199),('ማራቶን',0.9942078590393066) and using most_similar_cosmul()are:[('ጥሩነሽ',0.9994797110557556),('ቀነኒሳ',0.998009443283081) ('ሜትር',0.9978436231613159),('አልማዝ',0.9972312450408936),('ማራቶን',0.9971029758453369)]. This shows only the values are different but the words are still the same and occurring in the same context.

Table 5.5: Sample Terms with Similar and Non-similar Words

Terms	Vocabularies	
	Related words	**Non-related words**
አፍሪካ	‘ማጣሪያ', ‘ቻን', 'ሴካፋ', 'ዞን', 'ሩዋንዳ', 'ኡጋንዳ', 'ሱዳን', 'ምስራቅ', 'አስተናጋጅነት', 'ምድብ', 'ሞሮኮ', 'ናይጄሪያ', 'ኮንጎ','ዋንጫ', 'ግብጽ'	'ችግር', 'አሰራር', 'ጭምር', 'ባለሙያ', 'ሚገባ', 'ሙያተኛ', 'አሉ', 'አደረጃጀት', 'መፍትሄ', 'ክትትል', 'ሰው', 'ህግ', 'አካላት', 'ተቋም'
ኦሊምፒክ	'አትሌቲክ', 'ተወዳዳሪ', 'ገብረስላስ', 'ስላሴ', 'ሻለቃ', 'ገብረ', 'ማእከል','ተአምር', 'ማርያም', 'አቀፍ', 'ብስክሌት', 'ሪዮ', 'ቴክዋንዶ', 'ቶኪዮ', 'እግዚአብሄር'	'ተጨዋች', 'ክለብ', 'ቡድን', 'ሲዝ', 'ግጥሚያ', 'ዝውውር', 'ማንቸስተር', 'ቼንገር', 'ደጋፊ', 'አርሰናል', 'ተከላካይ', 'ጨዋታ',

ወርቅ	'ሜዳሊያ', 'ነሀስ', 'ሜትር', 'ማራቶን', 'ብር', 'ዲባባ', 'አያና', 'ቤጂንግ', 'አልማዝ', 'ጥሩነሽ', 'ሺህ', 'በለንደ', 'ምሩጽ', 'ለንደን', 'መሰናክል',	'ደንብ', 'መመሪያ', 'ውሳኔ', 'ተገቢ','መመርያ', 'ግልጽ', 'አደረጃጀት', 'ችግር', 'ህግ', 'አሰራር', 'ክፍተት', 'ምን', 'ሆነ', 'መፍትሄ', 'ጥያቄ',
ኦሮሚያ	'አማራ', 'ሶማሌ', 'ደቡብ','ታውቋል', 'መሀመድ', 'ትግራይ','ቴክዋንዶ', 'ሚያዝያ', 'ሀረሪ', 'ኢንተርናሽናል', 'ሰለሞ', 'ዩኒቨርሲቲ', 'ነሀሴ','ሀሙስ', 'ፖሊስ'	'ተጨዋች', 'ቬንገር', 'እኔ','ችግር', 'እምነት','ሊሆን', 'ዝውውር','ምን','አርሴናል','ብቃት', አሉ, 'ምክንያት', 'ሲዝ','ሳይሆን',
ሲቲ	'ዩናይትድ', 'ሊቨርፑል', 'ሻምፒዮንነት', 'ማንቸስተር','ማንችስተር', ‘ግጥሚያ','አጥቂ','ማድሪድ','እንግሊዝ', 'ነል','ሲዝን','አርሰናል','ቼልሲ', 'ተከላካይ'	'ኮሚቴ','አቶ','አሸብር','አስፈጻሚ','ፌዴሬሽ','ፕሬዚዳንት', 'ምርጫ', 'እጩ', 'ጉባኤ', 'ፌዴሬሽን','ሚኒስትር', 'አስመራጭ'

For the term አፍሪካ, አሊምፒክ, ወርቅ, ኦሮሚያ, and ሲቲ after splitting and searching from the model the top most similar words and totally non-related words can be retrieved. This shows for a given word there are words that are occurring on the same orientation. The cosine values which are approaching to -1 or zero are totally far in meaning and not considered for semantic information. For example, for the word አሊምፒክ, the following figure shows the top 10 similar and non-similar words.

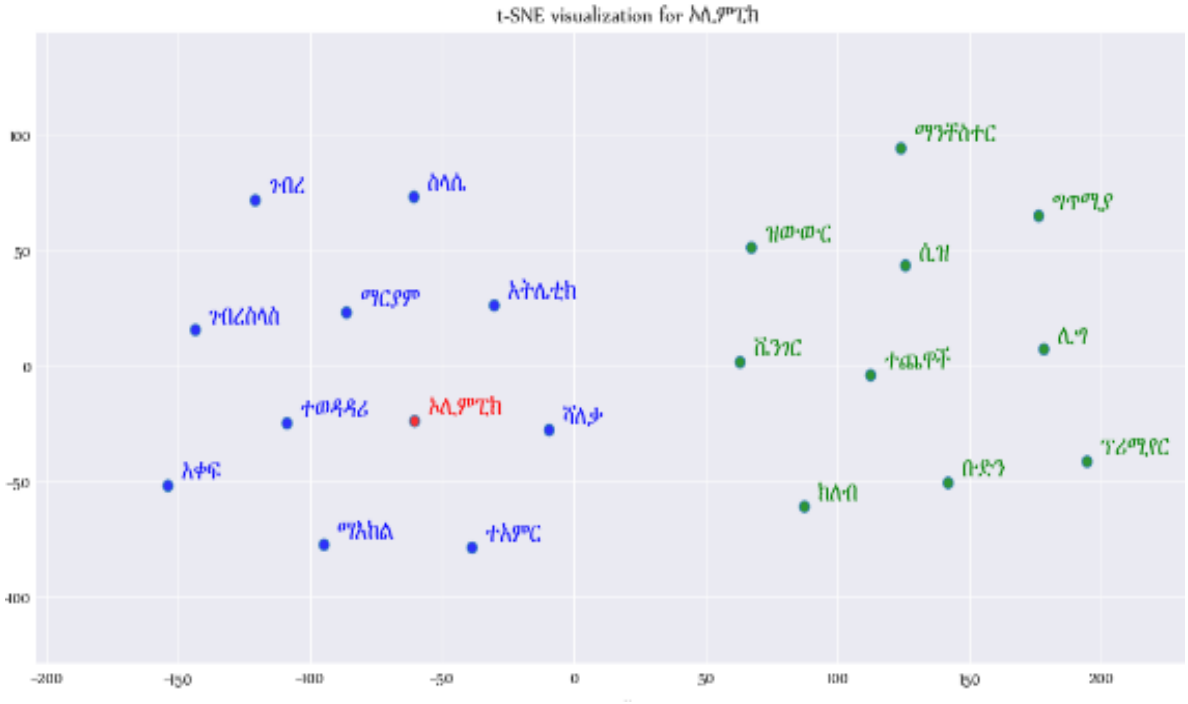

Figure 5.5: Sample Related and Non-related Words for አሊምፒክ

As it is shown in the above figure the word colored with red is the query word and words colored with blue are related words and green colored words are completely opposite words which are non-related words according to the vector distances.

Searching

After the system is successfully implemented in Word2Vec the users can directly search the semantically related words using the graphical user interface. The GUI is first designed by the Qt Designer and implemented in python. The user selects the domain and determines the top n number of words then by clicking the search button the results will be displayed on the text field.

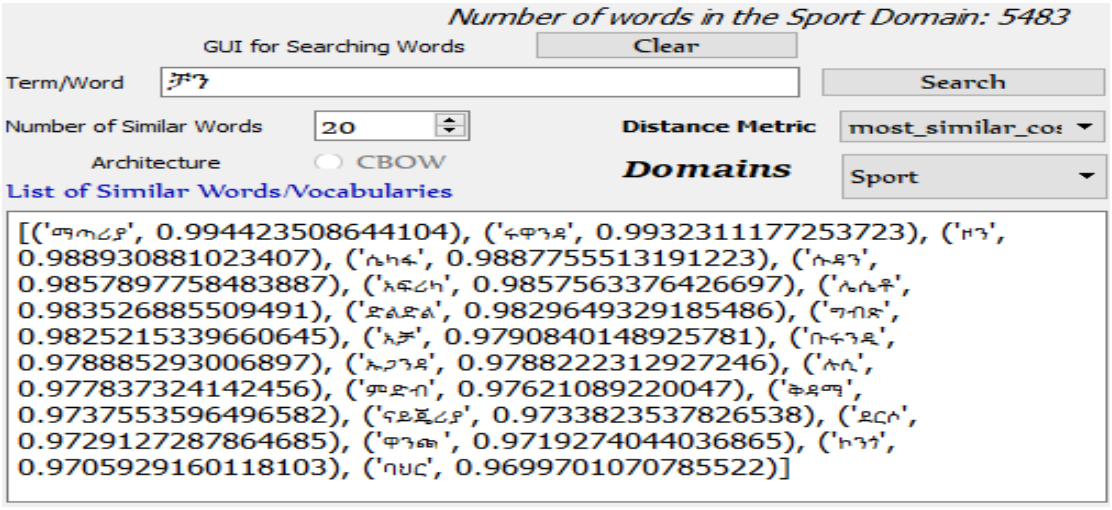

Figure 5.6: GUI for Semantic Vocabulary Searching Using Score

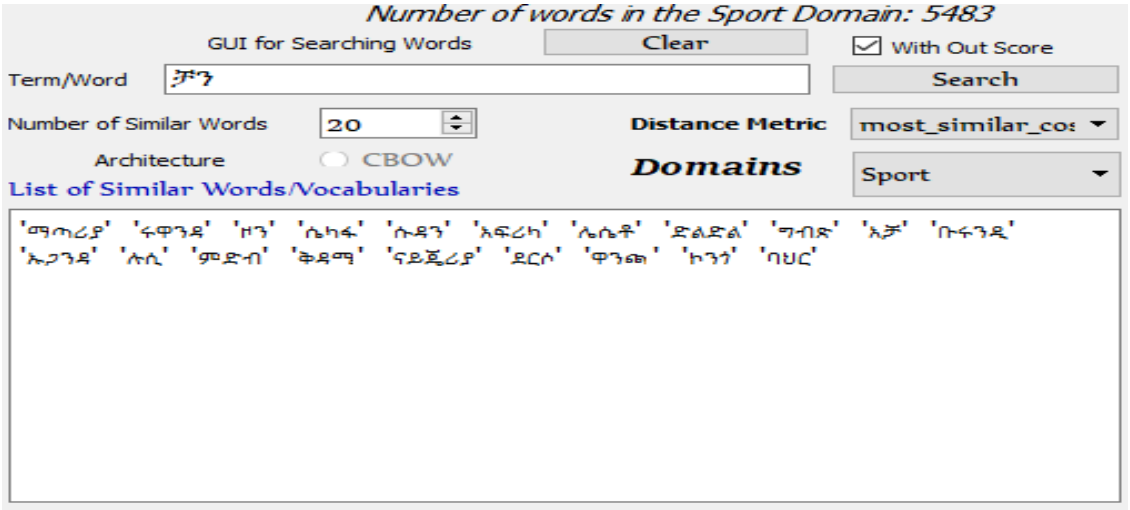

Figure 5.7: GUI for Semantic Vocabulary Searching without Score

5.7 Evaluation

Evaluation is a standard measuring technique that basically discusses or shows how much the research objective is achieved. It may be increasing the accuracy or efficiency of the system by comparing it with the previous researches worked in a domain or make resources accessible for a certain client. It depends on the research objective and idea. And, it can be expressed with or without numbers. For example, the intrinsic evaluation of unsupervised learning cannot directly be expressed using numbers rather than predicting certain contextual words based on the model is learned to predict the words surrounding the target word. The techniques for evaluating our system are discussed and implemented hereunder basically into two folds.

5.7.1 Word2Vec Model Evaluation

Intrinsic: - It implies measuring the performance of the system using the word's semantic relations. We have tried to show the semantic relation and implemented some of the intrinsic methods based on the following measures.

1. Analogical Measure

The notion behind analogy is to find a word m for a given word n so that m: n best resembles as a sample relationship w: w' or in a simple expression if a is to b as c is to d. For example the query term containing words like ["ስልክ መረጃ ነግል", "አሜሪካ መተግበሪያ ዶላር", "ኮምፒውተር አፕል ድምፅ"] and the semantic relation become like 'ስልክ' is to 'መረጃ' as 'ነግል' is to 'ቴክኖሎጅ', 'አሜሪካ' is to 'መተግበሪያ' as 'ዶላር' is to 'ስጋት' and 'ኮምፒውተር' is to 'አፕል' as 'ድምፅ' is to 'መልእክት'.

By applying this notion, we have prepared 14 sample questions to measure the word analog similarity. Comparatively, with the words vectorized with latent semantic analysis (LSA) for seven sample questions that are answered by 10 linguistic experts, 51.1% similarity value was obtained, but using Word2Vec 68.3% average score was recorded. This shows using Word2Vec embedding was increased the score by 17.2 which enhances efficiency for vector representation. The reason for decreasing results in LSA may be due to improper handling of stop words, the embedding parameter consideration and the way words are vectorized. Embedding based on context words is better than embedding based on bag of word count using LSA as it is shown in Annex C.

2. Relatedness

It has high cosine values between a set of words approaching 1. For example, the cosine similarity of ሲቲ and ዩናይትድ is 0.998855307698038 which means ሲቲ is highly related to the term ዩናይትድ than other words else in the vocabulary. So generally, Word2Vec by itself can be evaluated using intrinsic evaluation. It is also evaluated in the area of information retrieval using query expansion which can increase the number of relevant documents to be retrieved.

5.7.2 Evaluation Using Information Retrieval

Information retrieval using whoosh is an alternative to Apache Lucene which is a java implementation and it is a pure python library together with many term scoring algorithms that are basically used in document indexing and searching. Retrieval relevancy can be based on either ranked or unranked measures.

5.7.2.1 Based on Unranked Retrieval Relevance Measure

The unranked set of retrieval returns the relevant documents without considering the relevance order. Even if it has been the relevance measure before some years ago when the indexed data is huge and the users are looking for a result for their query it may be tedious to find the relevant documents. The researchers have developed the semantic thesaurus with latent semantic analysis [15] and they have tested in an information retrieval system that has obtained 73.34% recall. We have compared with their work and our system has been improved to 84.23% recall. It is more depicted in the following table 5.8.

Table 5.6: Precision and Recall for Document Searching Unranked Set of Retrieval

Queries	REL	Using Semantic Vocabulary						Without Using Semantic Vocabulary					
		RET	RET REL	NRET REL	RET NREL	R	P	RET	RET REL	NRET REL	RET NREL	R	P
ኤች አይ ቪ ኤድስ ከላከል ቆጣጠር መንገድ	10	55	9	1	46	0.9	0.2	40	9	1	31	0.9	0.22
ኢድ አልፈጥር በአል	10	43	9	1	34	0.9	0.21	29	8	2	21	0.8	0.27
ሁለገብ ገበሬ ህብርት ስር ማህበር	12	49	7	5	42	0.60	0.143	33	2	10	31	0.17	0.06
ኢትዮጵያ አየር መንገድ	11	168	10	1	158	0.91	0.1	84	10	1	74	0.91	0.12
ታላቅ ህዳሴ ግድብ	10	23	10	0	13	1	0.45	11	9	1	2	0.9	0.9
ጽድ ዘመቻ	10	39	9	1	30	0.9	0.231	7	7	3	0	0.7	1
ንጹህ መጠጥ ውሃ	10	18	9	1	9	0.9	0.6	5	5	5	0	0.5	1
ስር አጥ	10	71	9	1	62	0.9	0.12	37	8	2	29	0.8	0.21
ስቅል በአል	7	43	4	3	39	0.571	0.093	29	4	3	39	0.571	0.18
Average						0.8423	0.2386					0.69	0.44

Description

- ✓ REL – The number of relevant documents
- ✓ RET – Total number of retrieved documents
- ✓ RETREL - Documents retrieved and relevant
- ✓ NRETREL – Documents not retrieved but relevant
- ✓ RETNREL – Documents retrieved but not relevant
- ✓ P – Precision, R – Recall
- ✓ The formula for Recall= (relevant ∩ retrieved)/ (relevant) or TP/TP+FN
- ✓ , Precision= (relevant ∩ retrieved)/retrieved, or TP/TP+FP

We have implemented the information retrieval system with semantic vocabulary with 9 queries and a total of 90 relevant and 518 indexed documents expanded with the top 5 number of similar words. When increasing the number of expanded words, the probability of returning the relevant documents becomes increasing. Due to this the recall becomes increasing. Hence the retrieval system is based on an unordered list of relevance, when the indexed documents are large it may be difficult for the users to look for relevant documents. Precision becomes decreasing, this is expected due to the increasing number of retrieved documents, but what is suggested is using semantic vocabulary for information retrieval system, provides more searching results that are relevant to be gained and be accessible. Using semantic vocabulary as a query expansion we have achieved 84.23% recall, and 23.86% precision respectively. Without using semantic vocabulary 69% recall, and 44% precision were obtained respectively. Generally, a recall is never decreasoning while using semantic vocabulary and our system is better for recall oriented applications.

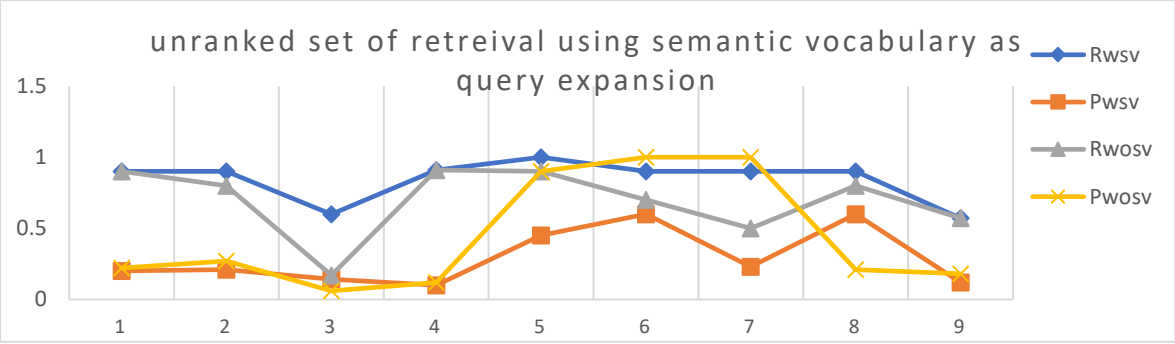

Figure 5.8: Information Retrieval Using Semantic Vocabulary Unranked Set of Retrieval

Description

Rwsv recall with semantic vocabulary

Rwosv recall without semantic vocabulary

Pwsv precision with semantic vocabulary

Pwosv precision without semantic vocabulary

5.7.2.2 Based on Ranked Retrieval Relevance Measure

Retrieving relevant documents are query-based in which the similarity of the query with the matched documents can be calculated and ranked, by which Google applies such types of techniques. User happiness is warranted while using the ranked retrieval. We have focused on retrieving the number of relevant documents based on the relevance of query terms. Using query expansion of semantic vocabulary there is a high probability for the relevant documents to be retrieved in descending order. We have implemented the three ranking function called Frequency, TF-IDF**,** and BM25F values. Firstly, the score of the query terms are calculated, then the relevant documents are retrieved in descending order via its ranks. For example, for the query term, **ሁለገብ ገበሬ ህብር ስራ ማህበር** using semantic vocabulary as a query expansion a total of seven relevant documents using BM25F metrics are matched and two relevant documents are matched without using query expansion. So, from saving the searching time point of view using query expansion it is advisable and initiative.

Table 5.7: Information Retrieval with Query Expansion Ranked Retrieval

input your queries that you want to search: "**ሁለገብ ገበሬ ህብር ስራ ማህበር**"	
Do you want query expansion (**QE**)? **Yes**	Do you want query expansion? **No**
How many similar words to expand? 5	topN=5
The expanded semantic words from the input **ሁለገብ ገበሬ ህብር ስራ ማህበር** are 'ቋረጥ', 'ሰፈራ', 'ጉልበት', 'ፍራፍሬ', 'ወስድ', 'ሁለገብ', 'ህበር', 'አርስ', 'ሚሊየን', 'ሰማር', 'በልጥ', 'ገበሬ', 'ማን', 'ሆኖ', 'ትክክል', 'ስራ', 'ቀበሌ', 'ቅረብ', 'ወጭ', 'አጥ', 'ሰብል', 'ማህበር'	No expanded semantic words. it is just the query word itself. input=**ሁለገብ ገበሬ ህብር ስራ ማህበር**
Search Results Found Total Number of Retrieved Documents: 49 Total Number of Indexed Documents: 518 The first five retrieved documents with score **Document_Rank: 1** Title General Domain Text Files Path: C:/Users/Pc/testing_data/102.txt Term Score 17.90287130424752 **Document_Rank: 2** Title General Domain Text Files	Search Results Found: Total Number of Retrieved Documents: 33 Total Number of Indexed Documents: 518 The first five retrieved documents with score **Document_Rank: 1** Title General Domain Text Files Path: C:/Users/Pc/testing_data/ 102.txt Term Score 9.45098542200743 **Document_Rank: 2** Title General Domain Text Files

Path: C:/Users/Pc/testing_data/130.txt Term Score 16.71196728412747 **Document_Rank: 3** Title General Domain Text Files Path: C:/Users/Pc/testing_data/ 99.txt Term Score 15.479970769652581 **Document_Rank: 4** Title General Domain Text Files Path: C:/Users/Pc/testing_data/ 92.txt Term Score 14.911731205520626 **Document_Rank: 5** Title General Domain Text Files Path: C:/Users/Pc/testing_data/ 94.txt Term Score 8.962728621348434 **Average_BM25F** is 14.793853836979327	Path: C:/Users/Pc/testing_data/2.txt Term Score 7.901016313469223 **Document_Rank: 3** Title General Domain Text Files Path: C:/Users/Pc/testing_data/ 92.txt Term Score 7.901016313469223 **Document_Rank: 4** Title General Domain Text Files Path: C:/Users/Pc/testing_data/gen (334).txt Term Score 7.2795174984541635 **Document_Rank: 5** Title General Domain Text Files Path: C:/Users/Pc/testing_data/gen (265).txt Term Score 6.734892836326503 **Average_BM25F** is 7.853485676745308

For reducing search complexity, we have determined the semantic words to be expanded and retrieved to the top five. Users can also determine the number of documents to be searched for the queries, but when increasing the number of documents, it becomes tedious and cumbersome for the searchers to refer to the relevant documents when using an unordered set of retrieval. The following table shows the relevant documents retrieved with the three basic query terms scoring.

Table 5.8: Term Scoring for Indexed Document Using Different Ranking Functions

Ranking Function	Users Query: input: = ሁለገብ ገበሬ ህብር ስራ ማህበር					
	Without query expansion			With query expansion		
	Avg Score	Retrieved Docs	Relevant Documents	Avg Score	Retrieved Docs	Relevant Documents
Frequency (avg)	1.00	10	2	3.6	10	3
TF-IDF (avg)	4.6	10	2	19.4	10	3
BM25F(avg)	6.8	10	2	11.54	10	6

We have implemented the query scoring algorithm with the first ten retrieved documents. As it is clearly shown in the table above even the value for term scoring is increased when using

TF-IDF, but from retrieving the relevant document point of view the BM25F function is the best. Using BM25F with ORGroup Boolean search retrieves the relevant documents in the high priority ranking orders. Due to this the mean average recall becomes increased because the relevant documents have a high probability of being retrieved as shown in table 5.9 below.

Table 5.9: Sample Recall and Precision at K Values for ሁለገብ ገበሬ ሀብር ሰራ ማህበር

Query		Precision at K Sample Ranks					Recall at K Sample Ranks				
		P@1	P@2	P@5	P@7	P@9	R@1	R@2	R@5	R@7	R@9
ኤች አይ ቪ ኤድስ መከላከል መቆጣጠር መንገድ	With semantic vocabulary	1	1	1	1	0.89	0.1	0.2	0.5	0.7	0.8
	Without Semantic vocabulary	0	0.5	0.8	0.71	0.8	0	0.1	0.4	0.5	0.7

It is usually known that the mean average recall and precision for ranked documents are calculated using randomly selected K values. When using semantic vocabulary as a query expansion 46% and 97.8% mean average recall and precision were recorded respectively. 34% and 56.2% mean average recall and precision was recorded without using semantic vocabulary.

Table 5.10: Mean Average Precision and Recall for Ranked Document Retrieval

Query No	Query List	Using Semantic Vocabulary		Without Using Semantic Vocabulary	
		Recall	Precision	Recall	Precision
1.	ኤች አይ ቪ ኤድስ መከላከል መቆጣጠር መንገድ	0.75	0.70	0.56	0.67
2.	ኢድ አልፈጥር በአል	0.70	0.43	0.60	0.9
3.	ሁለገብ ገበሬ ማህበር	0.69	0.841	0.11	0.72
4.	ኢትዮጵያ አየር መንገድ	0.671	0.85	0.45	0.60
5.	ታላቅ ህዳሴ ግድብ	0.71	0.87	0.65	0.92
6.	አካል ጉዳት	0.68	0.60	0.54	0.97
7.	አድዋ ድል በዓል	0.61	0.99	0.53	1
8.	አርብ አደር	0.55	1	0.31	0.32
9.	ጽድ ዘመቻ	0.78	0.834	0.40	1
10.	ንጹህ መጠጥ ውሃ	0.74	0.59	0.3	1
	Mean Average	**0.6881**	**0.7705**	**0.4451**	**0.81**

When the size of the input query increases the corresponding expanded query terms and the relevant documents have a higher probability of being retrieved. This enables the information retrieval system to retrieve the most relevant documents according to its rank in descending orders. This saves the searching time for the users while they are seeking the information they want. The mean average recall and precision are calculated depending on the number of retrieved documents via its rank. By implementing the ranked retrieval system expanded with the semantic vocabulary, we have achieved 68.81%, and 77.05% mean average recall, and precision respectively. Without semantic vocabulary, it was recorded 44.51%, and 81% mean average recall, and precision, respectively. We have suggested that semantic vocabulary with ranked information retrieval is better to use and implement.

Chapter Six: Conclusion, Contribution, and Future Work

6.1 Conclusion

Nowadays the need to use and access lexical resources to enhance information retrieval using query expansion become more fundamental in natural language including Amharic. The resources made available for query expansion may be either manually build or automatically created from large document collections. Contextual meaning extraction also becomes more essential these days. Creating resources manually is labor-intensive, and time-consuming when the dataset is huge. To solve such kinds of difficulties an automatic option has been proposed and using distributional semantics is the new approach to use and implement.

The proposed approach is implemented in the Word2Vec algorithm that could able to construct the semantic vocabulary and it is used for information retrieval system. To achieve our objectives, we have performed the natural language processing tasks and we have implemented the word-space modeling based on the fundamental parameters. We have manually crawled 8,759 Amharic documents from different domains and after smoothing the text corpus we have built the list of words that are ready for learning. Using the Word2Vec algorithm we have constructed a total of 44,497 lists of semantically related vocabulary from nine domains. But we can increase the number of words by decreasing the minimum occurrence count of less than five. Once the word vector is successfully learned and get trained it can be used for searching. To search the semantic vocabulary, we have built the graphical user interface so that the users can easily interact with the system and similar words will be retrieved based on their cosine orientation.

The incorrectly stemmed words and stop words affect the training as well as the word quality. Alphabets like ሸ, ዋ, ሀ are sometimes useful but not always been considered in the semantic vocabulary list. Our semantic vocabulary is better than the thesaurus constructed by Andargachew Mekonn using latent semantic analysis (LSA) with singular value decomposition (SVD) dimension reduction as the human linguists verified it with fourteen sample questions prepared for testing analog questions. Hence no WordNet developed for Amharic we have measured the word similarity based on the user prepared lists. 51.1% analogy similarity was achieved using LSA and 68.3% performance was obtained using word2vec. So neural word embedding using word2vec is better than other traditional embeddings. Performance of unsupervised learning is not directly measured which depends on the end goals of a certain task but possible to test based on the intrinsic evaluation and it was compromising.

Our semantic vocabulary is performed better for information retrieval by expanding the query terms with the top five most similar terms. It was tested based on a ranked and unranked set of retrieval. Even if the recall becomes relatively increasing when using semantic vocabulary as a query expansion based on ranked and unranked retrieval. Using unranked information retrieval 84.23% recall was achieved and this is increased by 10.89% recall than the semantic thesaurus built by Andargachew Makonnen. We conclude that ranked retrieval is better than unranked retrieval. Using ranked retrieval many relevant documents were retrieved in their priority relevance based on the ranks in descending order. When the indexed documents are too huge ranked retrieval could save searching time than an unranked retrieval system. By applying ranked retrieval 68.81% and 44.51% mean average recall was recorded using and without using semantic vocabulary as a query expansion respectively. It was improved by 24.3% recall.

As experimental results show the BM25F ranking function outperforms better than other ranking functions like frequency-based retrieval and relevant documents have a high degree of probability being retrieved.

6.2 Contributions

The main contribution of our work is summarized as follows.

- ✓ We can construct contextually similar Amharic semantic vocabulary using Word2Vec.
- ✓ We have integrated the semantic vocabulary into an information retrieval system and we could retrieve many relevant documents by expanding the query words into a set of its corresponding semantic words. The output of this system is a cluster of semantic words and which used as an input for information retrieval systems to expand the user's query.
- ✓ We proved that ranked retrieval can better perform search relevance than the standard unorder set of retrieval.
- ✓ Our own architecture

6.3 Future Work

This research work can be further extended and enhanced:

- ✓ Using the word-space of different domains by specifying the word relations and using parts of speech contextual WordNet can be automatically constructed.
- ✓ Using the web and by considering the parts of speech semantic vocabulary can be accessed online for any user like visual thesaurus developed for English.
- ✓ The information retrieval system can be enhanced by expanding the queries with semantic vocabulary combined with manually built WordNet.

References

[1] Chowdhury and G. Gobinda, "Natural language processing," *Annual review of information science and technology*, vol. 37, no. 1, pp. 51-89, 2003.

[2] K. Venelin, S. Maria and M. M. Antonia, "Comparing Distributional Semantics Models for identifying groups of semantically related words," *Procesamiento del Lenguaje Natural*, no. 57, pp. 109-116, 2016.

[3] "aurelieherbelot.net," [Online]. Available: http://aurelieherbelot.net/research/ distributional-semantics-intro/. [Accessed 02 Dec 2017].

[4] B. Gemma and H. Aurelie, "Formal Distributional Semantics:Introduction to the Special Issue," *Association for Computational Linguistics*, vol. 42, no. 4, pp. 619-635, 2016.

[5] B. Elia , K. T. Nam and B. Marco , "Multimodal Distributional Semantics," *Journal of Artificial Intelligence Research*, vol. 49, no. 2014, pp. 1-47, 2014.

[6] B. Romaric, R. Martin and C. Jean-Cédric, "Textual Similarities based on a Distributional Approach," in *Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on*, IEEE, 1999, pp. 180-184.

[7] L. Alessandro, "Will Distributional Semantics Ever Become Semantic?," in *7th International Global WordNet Conference*, Tartu, 2014.

[8] H. Abram, "An empirical study of semantic similarity in WordNet and Word2Vec," unpublished masters thesis, Orleans, 2014.

[9] A. Magnus, S. Maria, K. Shiho, R. Rafal, and A. Kenji, "Medical vocabulary mining using distributional semantics on Japanese patient blogs," in *6th International Symposium on Semantic Mining in Biomedicine,(SMBM), Aveiro, October 6-7*, Portugal, 2014.

[10] Rahman, M. a. Asker, L. a. Skeppstedt and Maria, "Proposing distributional semantics as a tool for medical vocabulary expansion," in *International Workshop on Embeddings and Semantics (IWES'15), 15 September*, Alicante, 2015.

[11] Enrique Henestroza Anguiano and Pascal Denis, "FreDist: Automatic construction of distributional thesauri for French," in *TALN-18{\`e}me conf{\'e}rence sur le traitement automatique des langues naturelles*, Le Chesnay Cedex, France, 2011, pp. 119-124.

[12] C. Vincent and K. Ewa, "Distributional Thesauri for Information Retrieval and vice versa," in *Language and Resource Conference*, LREC, 2016.

[13] Alelgn Tefera , "Automatic Construction of Amharic Semantic Networks(ASNet)," *Unpublished Masters Thesis, Department of Computer Science,Addis Ababa University*, 2013.

[14] Segid Hassen, "AMHARIC WORD SENSE DISAMBIGUATION USING WORDNET," *Unpublished Masters Thesis, Department of Computer Science,Addis Ababa University*, 2015.

[15] ANDARGACHEW MEKONNEN, "AUTOMATIC THESAURUS CONSTRUCTION FOR AMHARIC TEXT RETRIEVAL," *Unpublished Masters Thesis, Department of Computer Science,Addis Ababa University*, pp. 1-103, July 2009.

[16] "abyssinica.com," [Online]. Available: https://dictionary.abyssinica.com/amharic.aspx. [Accessed Sat Mar 17 2018].

[17] "www.readingrockets.org," [Online]. Available: http://www.readingrockets.org/article/teaching-vocabulary. [Accessed Tue Feb 6 2018].

[18] Bruni Alia, Khanh Nam, and Baroni Marco, "Multimodal Distributional Semantics," *Journal of Artificial Intelligence Research*, vol. 49, no. 2014, pp. 1-47, 2014.

[19] B. Gemma and H. Aurelie, "Formal Distributional Semantics:Introduction to the Special Issue," *Association for Computational Linguistics*, vol. 42, no. 4, pp. 619-635, 2016.

[20] D. Georgiana , P. Nghia The and B. Marco , "General estimation and evaluation of compositional distributional semantic models," in *Association for Computational Linguistics*, Sofia, Bulgaria, 2013.

[21] Martin, Dian I and Berry and Michael W, "Mathematical foundations behind latent semantic analysis," *Handbook of latent semantic analysis*, pp. 35-56, 2007.

[22] Peter W. Foltz, "Latent semantic analysis for text-based research," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 197-202, 1996.

[23] Wiemer-Hastings, Peter and Wiemer-Hastings, K and Graesser and A, "Latent semantic analysis," in *Proceedings of the 16th international joint conference on Artificial intelligence*, 2004, pp. 1-4.

[24] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth Conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1999, pp. 289-296.

[25] Ayush Jaiswal and Anunay Bhargava, "Explicit Semantic Analysis for Computing Semantic Relatedness of Biomedical Text," in *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)*, IEEE, 2014, pp. 929-934.

[26] Quechuan Zhang, "Data Jacket Retrieval Based on Explicit Semantic Analysis," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, IEEE, 2015, pp. 749-752.

[27] Jing LUO, Bo Meng, Xinhui Tu and Maofu Liu, "Concept-based Document Models using Explicit Semantic Analysis," in *2012 IEEE International Conference on Granular Computing*, IEEE, 2012, pp. 338-342.

[28] Philipp Sorg and Philipp Cimiano, "An Experimental Comparison of Explicit Semantic Analysis Implementations for Cross-Language Retrieval," in *International Conference on Application of Natural Language to Information Systems*, Springer, 2009, pp. 36-48.

[29] J. Yongxia, G. Heping, F. Chuanyi and L. Qiang, "Study on Text Classification Algorithm Based on Non-negative Matrix Factorization," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, IEEE, 2017, pp. 484-487.

[30] "Google Code Archive," [Online]. Available: https://code.google.com/archive/p/word2vec/. [Accessed Mon Feb 12 2018].

[31] Dyah Rahmawati and Masayu Leylia Khodra, "Word2vec semantic representation in multilabel classification for Indonesian news article," in *2016 International Conference On Advanced Informatics: Concepts, Theory, And Application (ICAICTA)*, IEEE, 2016, pp. 1-6.

[32] Ronghui Ju, Pan Zhou, Cheng Hua Li and Lijun Liu, "An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, IEEE, 2015, pp. 2276-2283.

[33] Jey Han Lau and Timothy Baldwin, "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation," *arXiv preprint arXiv:1607.05368*, 2016.

[34] C. Michal and J. Karel, "Comparing semantic models for evaluating automatic document summarization," in *International Conference on Text, Speech, and Dialogue*, Springer, 2015, pp. 252-260.

[35] P. Juan, D. Posadas, G. A. Helena and S. Grigori , "Application of the distributed document representation in the authorship attribution task for small corpora," *Soft Computing*, vol. 21, no. 3, pp. 627-639, 2017.

[36] Quoc Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31 st International Conference on Machine*, Beijing, China, 2014.

[37] Ruqing Zhang, Jiafeng Guo, Yanyan Lan and Jun Xu& Xue, "GENERATIVE PARAGRAPH VECTOR," in *China Conference on Information Retrieval*, Springer, 2018, pp. 105-118.

[38] J. Nicholas and C. W. Bruce , "Information filtering and information retrieval: Two sides of the same coin?," *Communications of the ACM*, vol. 35, pp. 1-10, Dec 1992.

[39] "http://dataconomy.com," [Online]. Available: http://dataconomy.com/2015/04/implementing-the-five-most-popular-similarity-measures-in-python/. [Accessed Fri Feb 17 2018].

[40] Anna Huang, "Similarity measures for text document clustering," in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, 2008, pp. 49-56.

[41] P. K. a. A. S. a. K. M. A. Verma, "Opinion mining considering roman words using Jaccard similarity algorithm based on clustering," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, 2017, pp. 1-4.

[42] Curran, James R and Moens, Marc, "Improvements in Automatic Thesaurus Extraction," in *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9*, Association for Computational Linguistics, 2002, pp. 59-66.

[43] V. Claveau and E. Kijak, "Distributional Thesauri for Information Retrieval and vice versa," in *Language and Resource Conference*, LREC, 2016.

[44] Maria Skeppstedt, Magnus Ahltorp, and Aron Henrikss, "Vocabulary expansion by semantic extraction of medical terms," in *Proceedings of the Symposium on Languages in Biology and Medicine (LBM). Database Center for Life Science*, 2013.

[45] Nega Alemayehu and Peter Willet, "stemming of Amharic words for information retrieval," *Literary and linguistic computing,* vol. 17, pp. 1-17, 2002.

[46] S. Weiwei and U. Hans, "Capturing Paradigmatic and Syntagmatic Lexical Relations:Towards Accurate Chinese Part-of-Speech Tagging," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Republic of Korea, 2012.

[47] "omniglot the online encyclopidia of writing and language system," [Online]. Available: https://www.omniglot.com/writing/amharic.htm. [Accessed Fri Mar 17 2018].

[48] Magnus Sahlgren, "the Word-Space Model Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector space," *Unpublished PhD Dissertation, Department of Linguistics,Stockholm University*, 2006.

Annexes

Annex A Sample Stop Word

ሁሉ	ላይ	ተጨማሪ	እባክሽ	ወይም
ሁሉም	በተለይ	ትናንት	እባክዎ	ወደፊት
ሁሉንም	ተመለከተ	ትናንትና	እና	ውጪ
ሁል	በተመሳሳይ	ነበረ	እንደ	ዉስጥ
ሆነ	በታች	ነበረች	እንደገና	የሚገኙ
ሆኑ	በኩል	ነበሩ	እንዲሁም	የሚገኝ
ሆኖም	በውስጥ	ነው	እንጂ	የሰሞኑ
ኋላ	በጣም	ነይ	እኛ	የተለያየ
ላይ	በፊት	ነገር	እያንዳንዱ	የተለያዩ
ሌላ	በሆላ	ነገሮች	እያንዳንዳችዉ	የታች
ሌሎች	ቢሆን	ናት	እያንዳንደ	የዉስጥ
ልዩ	ቢቢሲ	ናቸው	ከ	የጋራ
መሆኑ	ብለዋል	አሁን	ከኋላ	ይህ
ማለቱ	ብቻ	አለ	ከላይ	ደግሞ
ማለት	ብዙ	አቶ	ከመካከል	ድረስ
ማን	ብዛት	እሱ	ከሰሞኑ	ጋራ
ማንም	ተለያየ	እስከ	ከታች	ጋር
ማድረግ	ተለያዩ	እሷ	ከውስጥ	ግን
ሲሆን	ተባለ	እባኩህ	ከጋራ	ጥቂት
ሲል	ተገለጸ	እባኩሽ	ከፊት	ፊት
ስለ	ተገልጹል	እባኩዎ	ወዘተ	
በኋላ	ተጨማሪ	እባክህ	ወይ	

Annex B Sample Compound Words

Politics	Business	Sport	Art	Law	Health
ሸዋ ሮቢት	ድህረ ምረቃ	ክብረ ወሰን	ድረ ገጽ	ርእዮተ አለም	ፀረ ተህዋስያን
ማረሚያ ቤት	ግብረ ሀይል	ግብረ ሀይል	ንጉሠ ነገስት	አውደ ርእይ	ስንፈተ ወሲብ
ኣቃቤ ህግ	ክብረ በአል	ግብረ ሰናይ	አፈ ጉባዔ	ህገ ወጥ	ንቅለ ተከላ
ፍርድ ቤት	ፀረ አረም	እግር ኳስ	ዓመተ ምህረት	አርሶ አደር	ስነ አእምሮ
ፅህፈት ቤት	ቅርፃ ቅርፅ	እጅ ኳስ	ኪነ ጥበብ	ቤተ ዘመድ	ስነ ልቦና
ምክር ቤት		መረብ ኳስ	ጽንሰ ሀሳብ	ቁም ነገር	ሀፍረተ ገላ
ፕብሊክ ሰርቪስ		አለም አቀፍ	ዓለም አቀፍ	ግብረ መልስ	ስርአተ ፆታ
መርሃ ግብር		ገብረ ስላሴ	ቅድመ ታሪክ	ግብረ ስጋ	ስርአተ ምግብ
መድበለ ፓርቲ		ማንቸስተር ዩናይትድ	እሰጣ ገባ	ቃለ ጉባዔ	ስርአተ ልመት
ነፍስ ወከፍ		ማንቸስተር ሲቲ	ወዘተ ወዘተረፈ	ቃለ መጠይቅ	ግብረ ሰዶም
ግብር ከፋይ		ቅዱስ ጊዮርጊስ	ሥነ ስርዓት	ግምጃ ቤት	
ክፍለ ከተማ		ረኧም ርቀት	'ፍትሐ ብሔር	ህገ መንግስት	
አርብቶ አደር		አጭር ርቀት	ፍትሐ ነገስት	ሽንት ቤት	
ሊቃነ መናብርት		ሪያል ማድሪድ	ሕገ መንግስት	ሊቀ መንበር	
ዘርፈ ብዜራዊ		ጽህፈት ቤት	ሰበር ዜና	ሕገ ወጥ	
ድህረ ምርጫ			ሥነ ስርዓት	ጨርቃ ጨርቅ	
መራሂተ መንግስት			ቤተ መንግስት		
ንቃተ ህሊና			ዓይነ ስውር		
			ዓይነ ምድር		
			ዓይነ ብርሐን		
			ርዕሰ መስተዳድር		
			ርዕሰ መምህር		
			ኮማንድ ፖስት		
			ንጥረ ነገር		
			ስነ ባህርይ		

Annex C Word Analogy Prediction using Word2vec and LSA

No	Similar words pair using (LSA)	Total Score	Mean Score	Similar words pair using (Word2Vec)	Total Score	Mean Score	Number of Linguists
1.	(በግ,ሚዛን)	11	1.1	(በግ, አውራ)	34	3.4	10
2.	(በግ,ሚቃጠል)	14	1.4	(በግ, እንስት)	38	3.8	10
3.	(አንበሳ, ደቦል)	43	4.3	(አንበሳ, እንስሳ)	31	3.1	10
4.	(ታቦት, አቢዳር)	12	1.2	(ታቦት, አምድ)	29	2.9	10
5.	(ፈረስ, ዝንጀር)	19	1.9	(ፈረስ, ጦር)	32	3.2	10
6.	(ኪዳን, ቃል)	39	3.9	(ኪዳን, ተናገሩ)	35	3.5	10
7.	(እንጨት,ጥድ)	41	4.1	(እንጨት, ዝግባ)	40	4.0	10
	Total	179	17.9		239	23.9	
		Average	**51.1%**			**68.3%**	

Annex D Amharic Alphabets Normalized into One Consistent Letter

Alphabet	Normalized Words
ሃ, ኀ, ኃ, ሐ, ሓ	ሀ
ሑ, ኁ	ሁ
ሒ, ኂ	ሂ
ሔ, ኄ	ሄ
ሕ, ኅ	ህ
ሖ, ኆ	ሆ
ሠ	ሰ
ሡ	ሱ
ሢ	ሲ
ሣ	ሳ
ሤ	ሴ
ሥ	ስ
ሦ	ሶ
አ, ዐ, ዓ	አ
ዑ	ኡ
ዒ	ኢ
ዓ	ኤ
ዕ	እ
ዖ	ኦ
ጸ	ፀ
ጹ	ፁ
ጺ	ፂ
ጻ	ፃ
ጼ	ፄ
ጽ	ፅ
ጾ	ፆ
ጐ	ጎ

Annex E Sample Questions Answered by the Linguists

Score

1=Far

2=Poor

3=Good

4=Very Good

5=Excellent

Name: _____

Education Level: _____

Occupation: _____

Answer the following questions based on the associated word score?

1. The word **በግ** is associated with **ሚዛን** with a score? (LSA)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
2. The word **በግ** is associated with **ሚቃጠል** with a score? (LSA)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
3. The word **በግ** is associated with **አውራ** with a score? (Word2Vec)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
4. The word **በግ** is associated with **እንስት** with a score? (Word2Vec)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
5. The word **አንበሳ** is associated with **ደቦል** with a score? (LSA)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
6. The word **አንበሳ** is associated with **እንስሳ** with a score? (Word2Vec)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
7. The word **ታቦት** is associated with **አቢዳር** with a score? (LSA)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
8. The word **ታቦት** is associated with **አምድ** with a score? (Word2Vec)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
9. The word **ፈረስ** is associated with **ዝንጅር** with a score? (LSA)
   A. 1 D. 4
   B. 2 E. 5
   C. 3
10. The word **ፈረስ** is associated with **ጦር** with a score? (Word2Vec)
    A. 1 D. 4
    B. 2 E. 5
    C. 3
11. The word **ኪዳን** is associated with **ቃል** with a score? (LSA)
    A. 1 D. 4
    B. 2 E. 5
    C. 3
12. The word **ኪዳን** is associated with **ተናገሩ** with a score? (Word2Vec
    A. 1 D. 4
    B. 2 E. 5
    C. 3
13. The word **እንጨት** is associated with **ጥድ** with a score? (LSA)
    A. 1 D. 4
    B. 2 E. 5
    C. 3
14. The word **እንጨት** is associated with **ዝግባ** with a score? (Word2Vec)
    A. 1 D. 4
    B. 2 E. 5

Sample Code about Word2Vec Vectorization

```
import sys, os

from os import listdir

from gensim.models import Word2Vec

from multiprocessing import cpu_count

input_stop_list=open("F:/class AAU/research
dsm/Stop_List/my_list_of_stop_words.txt",encoding='utf-8')

input_stop_list=input_stop_list.readlines()

s=[x.split() for x in input_stop_list]

class MySentences(object):

    def __init__(self, dirname):

        self.dirname = dirname

    def __iter__(self):

        for fname in os.listdir(self.dirname):

            for line in open(os.path.join(self.dirname, fname),encoding='utf-8'):

                i=0

                for i in range(len(s)):

                    files=line.replace(str(s[i]),"")

                yield files.split()

file_dir = MySentences('C:/Users/Pc/sport_stemmed')  # a memory-friendly iterator

model_bible =
Word2Vec(file_dir,size=300,alpha=0.025,window=10,min_count=5,sample=1e-
4,workers=cpu_count(),min_alpha=0.0001,sg=0,hs=0,negative=25,cbow_mean=1,sorted_voc
ab=1,iter=15)

model_bible.wv.save_word2vec_format('model_sport.vec',binary=False)

model_bible.train(file_dir,total_examples=model_bible.corpus_count,epochs=25)

model_bible.init_sims(replace=True) #vector normalization
```

Sample Code about Vocabulary Construction

```
from gensim.models import Word2Vec

from gensim.models.keyedvectors import KeyedVectors

w2v_sport =KeyedVectors.load_word2vec_format('C:/Users/Pc/
model_sport.vec',encoding='utf-8',unicode_errors='strict')

similar_words = {search_term:[item[0] for item in
w2v_sport.wv.most_similar_cosmul([search_term],topn=15)]

            for search_term in y} #words are compared with each other inside the model.

sp = open("F:/class AAU/research dsm/vocabulary/new one/semantic vocabulary
offline/w2v_sport.txt","w",encoding="utf-8")

sp.write(similar_words)

sp.close()
```

Sample Code about Vocabulary Searching

```
import sys, PyQt5, re

from PyQt5.QtWidgets import QApplication,QDialog,QMainWindow

from PyQt5.uic import loadUi

from gensim.models.keyedvectors import KeyedVectors

model_sport=KeyedVectors.load_word2vec_format('C:/Users/Pc/model_sport.vec',encoding
='u f-8',unicode_errors='strict')  #unicode_errors='ignore'

model_sport.init_sims(replace=True)

sp =sorted(model_sport.wv.vocab.keys())

class ui_class(QMainWindow):
   def __init__(self):
      super(ui_class,self).__init__()
      loadUi('C:/Users/Pc/Desktop/python_GUI/qt_sample.ui',self)
      self.setWindowTitle("qt_desine sample gui searching")
      self.pushButton_1.clicked.connect(self.on_pushButton_clicked)
   def clear_buton(self):
      self.textBrowser_1.setText("")
      self.lineEdit_1.setText("")
   def without_score(wordList):
```

```
    Expanded_words=str(wordList)
    Expanded_words = re.sub(r"[@,',[,],?,\,.,$,%,_,/,]", "", Expanded_words)
    Expanded_words =  re.sub(r'[0-9]+',"",Expanded_words)
    return Expanded_words
     if dom_text == 'Sport':
       if (words in list(model_sport.vocab)):
          self.label_5.setText("Number of words in the Sport Domain: "+str(len(sp)))
          if sim_metric=='most_similar_cosmul':
             w2vec_list=model_sport.wv.most_similar_cosmul(words,topn=topn_words)
             listofwords=ui_class.without_score(w2vec_list)
          else:
             w2vec_list=model_sport.wv.most_similar(words,topn=topn_words)
             listofwords=ui_class.without_score(w2vec_list)
          self.textBrowser_1.setText((listofwords))
       else:
          self.label_5.setText(" The Word "+words+" is not in "+dom_text+" domain of the
model Model:")
    else:
       self.label_5.setText("Invalid Domain:")
app=QApplication(sys.argv)
widget=ui_class()
widget.show()
sys.exit(app.exec_)
```

Sample Code about Stop Word Filtering

```
from string import punctuation
from operator import itemgetter
N = 2500
words = {}
words_gen = (word.strip(punctuation).lower() for line in open("C:/Users/Pc/Desktop/amharic
bible/biblefile.txt",encoding='utf-8')
for word in line.split())
```

```
for word in words_gen:
    words[word] = words.get(word, 0) + 1
    top_words = sorted(words.items(), key=itemgetter(1), reverse=True)[:N]
for word, frequency in top_words:
    print("%s: %d" % (word, frequency))
```

Sample Code about Spell Inconsistency and Normalization

```
new_contents=re.sub(r"[ሐ,ኀ,ሓ,ኃ,ሃ]","ሀ",texffile)
new_contents= re.sub(r"[ሑ,ኁ]" , "ሁ",new_contents) #normalizing ሐ and ኀ alphabet to ሀ
new_contents= re.sub(r"[ሒ,ኂ]", "ሂ",new_contents)
new_contents= re.sub(r"[ሔ,ኄ]", "ሄ",new_contents)
new_contents= re.sub(r"[ሕ,ኅ]", "ህ",new_contents)
new_contents= re.sub(r"[ሖ,ኆ]", "ሆ",new_contents)
new_contents=re.sub('ዶ/ር','ዶክተር',new_contents)
new_contents=re.sub('አ/አ','አዲስ-አበባ',new_contents)
new_contents=re.sub('ተ/መ/ድ','የተመደ',new_contents)
new_contents=re.sub('እ/ኤ/አ','እኤአ',new_contents)
new_contents=re.sub('ኢ/ዜ/አ','ኢዜአ',new_contents)
new_contents=re.sub('ኤፍ/ቢ/ሲ','ኤፍቢሲ',new_contents)
```

Sample Code about Document Indexing

```
import os,re
from whoosh.index import create_in
from whoosh.fields import Schema, TEXT, ID
from whoosh.writing import AsyncWriter
from datetime import datetime, timedelta
from whoosh import fields, index
import sys
def createSearchableData(root):
    Stop_word = open("F:/class AAU/research dsm/Stop_List/my_list_of_stop_words.txt",encoding='utf-8')
    sp=[x.split() for x in Stop_word]
```

```
    schema =
Schema(title=TEXT(stored=True),path=ID(stored=True,unique=True),content=TEXT,textdat
a=TEXT(stored=True),date=fields.DATETIME)
    if not os.path.exists("indexdir_bible"):
        os.mkdir("indexdir_bible")
    ix = create_in("indexdir_bible",schema)
    writer = AsyncWriter(ix)
    y=[]
    filepaths = [os.path.join(root,i) for i in os.listdir(root)]
    for path in filepaths:
        i=1
        fp = open(path,'r',encoding='utf-8')
        text = fp.read()
        text = text.strip()
        text=re.sub([x.split() for x in Stop_word],",text)
        text=text
        writer.add_document(title="Religeous domain text file",
path=path,content=text,textdata=text,date=datetime.utcnow())
        ++i
        print(text)
        fp.close()
    writer.commit()
root ='C:/Users/Pc/bible/'
createSearchableData(root)
```

Sample Code about Document Searching using Query Expansion

```
from __future__ import division
from whoosh.qparser import QueryParser,OrGroup
from whoosh import scoring,qparser
from whoosh.index import open_dir
from whoosh.fields import *
from whoosh.scoring import *
from whoosh.qparser import *
```

```
from whoosh.query import *
from gensim.models.keyedvectors import KeyedVectors
import re,l3
from whoosh import index
from whoosh.qparser import QueryParser
from whoosh.qparser.dateparse import DateParserPlugin
Average_BM25F=0
start = time.time()
model_sport =KeyedVectors.load_word2vec_format('C:/Users/Pc/model_sport.vec',encoding='utf-8',unicode_errors='strict')  #unicode_errors='ignore'
model_sport.init_sims(replace=True)
y=list(model_sport.vocab)
query_not_in_model=[]
def get_expanded_query_w2v(model, q0, k):
    og = qparser.OrGroup.factory(0.9)
    qe = []
    for word in q0.split(' '):
        if word in y:
            expanded_words = [pair[0] for pair in model.most_similar(word)[:k]]
            expanded_words.append(word)
            qe.append(expanded_words)
        else:
            query_not_in_model.append(word)
            continue
    return qe
print("\nThe words "+query_not_in_model+ " are not in the model:\n")
stemmed=[]
text="F:/query.txt"
text2="F:/query2.txt"
q1 = open(text,"w",encoding="utf-8")
query=input("Enter the query you want to search: any query\n")
query=query.replace('r[0-9]+',")
query=query.replace('*[\:\?!][\"\)\]]* *',")
query=query.split()
```

```
for x in query:
    if len(x)>1:
        q1.write(x+" ")
q1.close()
l3.anal_file('am',text,text2)
with open(text2,encoding="utf8") as f:
 for line in f:
   parts = line.rstrip('\n').split(" ")
   i=0
   for x in parts:
     if x=='stem:':
       stemmed.append(parts[i+1])
     i=i+1
unique_words=[]
for k in stemmed:
    if k not in unique_words:
        unique_words.append(k)
print("uniqe words from the input query \n",unique_words)
words=str(unique_words)
input_topn=int(input('Enter topN number of similar words to be searched :\n'))
input_topn=str(input_topn)
print("Select your domain you can use the following Hints:")
print("#===============Sport=S or s==============#")
input_Domain=input()
if (input_Domain=="S" or input_Domain=="s"):
    model_sport=model_sportt
expanded_words_list=str(get_expanded_query_w2v(model_sport,words,input_topn))
unique_words=[]
Expanded_words = expanded_words_list
ixreader = open_dir("indexdir")
 earcher=ixreader.searcher()
with ixreader.searcher(weighting=scoring.BM25F(B=0.75,K=1.5)) as searcher:
    doc_count = str(searcher.doc_count_all()) #document count
    doc_num = searcher.document_number() #document number
```

```
    yes=input("do you want query expansion ?")
    if(yes=="y" or yes=="Y"):
        print("the expanded semantic words from the  input ",words," are \n",Expanded_words)
        query = QueryParser("content", ixreader.schema,group=OrGroup).parse(u(Expanded_words))
        text=query
        results = searcher.search(query,limit=input_topn)
    else:
        query = QueryParser("content",
ixreader.schema,termclass=Variations,group=OrGroup).parse(u(words))
        results = searcher.search(query,limit=input_topn)
        print("Without query expansion is the query itself: ",words)
    print("Search Results Found")
    a=int(format(len(results)))
    print("Total Number of Relevant Documents : "+str(a))
    print("Total Number of Indexed Documents : "+doc_count)
    print("{} files matched: ".format(len(results)))
    print("Total Number of Relevant and Retrieved Documents : "+str(input_topn))
    for i in range(topN):
        rank_score = str(results[i].score)
        print("Document_Rank: ",results[i].rank+1,"\n","Title",results[i]['title'],"\n",
"Path:",results[i]['path'],"\n","Term Score "+str(results[i].score),"\n")
        Average_Frequency=(Average_Frequency)+(results[i].score)
        i=i+1
Average_Frequency=Average_Frequency/input_topn
print("Average_BM25F is",Average_Frequency)
```

Signed Declaration Sheet

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

Declared by:

Name: Berihun Getnet Akalu

Signature: _____

Date: October , 2019

Confirmed by advisor:

Name: Yaregal Assabie (PhD)

Signature: _____

Date: October , 2019