ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Concept-based Amharic Documents Similarity (CADS)

By

Addisalem Abera Wordofa

A THESIS SUBMITED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDISABABA UNIVERSITY IN PARTIAL FULFILMENTOF THE REQUIREMENT FORTHE DEGREE OF MASTER OF SCINECE IN COMPUTER SCIENCE

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Concept-based Amharic Documents Similarity (CADS) Measure

By

Addisalem Abera Wordofa

Advisor: Fekade Getahun (Ph.D)

Signature of the Board of Examiners for Approval

Name	Signature
1. Dr. Fekade Getahun, Advisor	
2. Dr. Yaregal Assabie, Examiner	
3. _____	_____

December, 2013

DEDICATED TO:

My Family

Acknowledgments

First and foremost, thanks to God for his support and being with me throughout my life.

Thanks to my advisor Dr. Fekade Getahun. This thesis would not be possible without your supervision, support, inspiration and encouragements during the entire process of working on this research.

I wish to express my gratitude to Ethiopian Languages Study and Research Center for writing Amharic Dictionary. The Dictionary used as a resource during developing Amharic WordNet.

I would like to thank my sweet friends specially, Genet Assefa, Rakeb Besiro, Abeba Ibrahim, Tofik Jemal, Birhanu Mengiste and Desalegn Abebaw for their help on reviewing the paper, encoding data and sharing knowledge.

Finally, I want to thank all the people who have contributed in one way or another on this thesis work.

List of Tables

List of Algorithms

List of Figures

List of Appendixes

Acronyms

Artificial Intelligence	AI
AmhWordNet	Amharic WordNet
CADS	Concept-based Amharic Document Similarity
CSM	Concept-based Similarity Measure
CTE	Concept Tree Extraction
DS	Document Similarity
LCS	Longest Common Subsequence
LSA	Latent Semantic Analysis
NLP	Natural Language Processing
PMI	Pointwise Mutual Information
POS	Part of Speech
SDSM	Semantic Document Similarity Measures
SS	Sentence Similarity
SVD	Singular Value Decomposition
SWS	Semantic Word Similarity
VSM	Vector space Model
WLM	Wikipedia Link-based Measure
WSD	Word Sense Disambiguation
WIC	Walta Information Center

Abstract

Similarity measure has significance in the area of NLP applications such as search engine, information extraction and document classification. These NLP applications are implemented in Amharic language. However, most of them rely on simple matching techniques or probabilistic method to measure similarity. These approaches do not always accurately capture conceptual relatedness as measured by humans. Some of the researches try to consider semantic nature of a document without handling ambiguity of words. In this research, we proposed Concept-based Amharic Document Similarity (CADS) by building AmhWordNet.

The objective of this research is to implement effective similarity measure of documents by considering issues like polysemy, synonymy and semantic relationship between words. The main components of the proposed system (CADS) are AmhWordNet and Concept-based Similarity Measure (CSM). CSM consists of Word Sense Disambiguation (WSD), Concept Tree Extraction and Semantic Similarity Measure modules.

The AmhWordNet is used as input during concept tree extraction and to implement WSD module. The extracted concept tree together with WSD module helps to find the semantic similarity between words. The output of word similarity is used to compute sentence similarity. Finally document similarity is computed based on sentence similarities.

The performance of CADS is evaluated using precision, recall and F-measure evaluation metrics. CADS without WSD (CADSWoWSD), Pointwise Mutual Information (PMI), Jaccard and Cosine similarity measures are implemented so that comparison between the five systems is done. According to the result we get from the experiment we conducted, the proposed system has better performance than the existing ones.

Key Words: Word Sense Disambiguation, Concept Tree Extraction, Amharic WordNet, Concept-based Similarity Measure.

Table of Contents

CHAPTER ONE

INTRODUCTION

1.1 Background

Nowadays, researchers are highly focusing on natural language processing (NLP). NLP is the process of analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications[**1**]. Paraphrasing input text document, translating a text document to another language, answering questions about contents of text documents, and drawing inferences from the text documents are among the goals of NLP. To achieve these goals, there is a need to compute the similarity between documents, which is crucial to solve the problem of discovering and organizing documents associated with similar property.

The concept of computing document similarity was begun in 1970s [**2**]. Salton & Lesk 1971[2], Dillon & Gray, 1983 [3]; Fagan, 1989 [4]; Fuhr, 1989 [5]; Griffiths, Robinson, & Willet, 1984 [6] and Sparck, 1971[7] are some of researchers who had been participated in the area of computing document similarity. Over the years, different approaches have been introduced to measure similarity between documents. Vector Space Model (VSM) [**8**] is the most common type of statistical approaches to measure similarity between documents represented with vectors. The other statistical approach is Latent Semantic Analysis (LSA) [**9**], which is designed to reduce vector space size (which is a problem in VSM) using singular value decomposition approach. Knowledge-based is another approach to measure document similarity [**10**]. It uses resources like WordNet to solve synonym, polysemous and WSD problems that are not considered in statistical approach.

Document similarity has been used as prerequisite in various areas like Information Retrieval and Text Clustering [**2**,**11**]. There are researches conducted in the area of document similarity in languages other than English. For instance, in 2002, Pouliquen [**12**] came up with an approach to calculate the semantic similarity of documents written in the same or different languages. In

research, the similarity computation was done once the document is represented in a language-independent format using descriptor terms of the multilingual thesaurus EUROVOC, and then calculating the distance between these representations.

Considering the local language, Amharic, very few researches have been conducted in the area of information retrieval and document classification. In 2002, text retrieval using self-organized document map was done by Mulugeta Baye [**13**]. Later in 2003, Amharic text retrieval using latent semantic indexing was implemented by Tewodros Hailemeskel [**14**]. Researchers have been conducted to implement cross lingual information retrieval using Amharic and other languages (English and French). Some of those researches are Amharic-English Information Retrieval [**15**] and Dictionary-based Amharic-French Information retrieval [**16**] which are done by Atelach Alemu, Argaw and Lars Asker in the year 2004 and 2006. Automatic categorization of Amharic news text using machine learning approach by Surafel Teklu in 2003[**17**] is one of the earliest researches which was done on the area of document classification to investigate the application of machine learning techniques in automatic categorization of Amharic news items. Researchers in [**18**] also worked on classifying Amharic web news aimed at compiling an Amharic corpus from the Web and automatically categorizing the texts. On the research, ranking of documents was performed using the cosine similarity measure which lack considering semantic of text. A research on concept-based automatic Amharic document categorization is conducted in 2009 [**19**]. This research lacks identifying the correct meaning of a given word (i.e. word sense disambiguation) and references list of single terms/words as document representative.

All these researches have contributed to the area of NLP to implement it in local language - Amharic. But those implementations of information retrieval and classification do not consider the semantics of the contents of documents, i.e. documents that are similar in their concept but different in their word representation are not considered or disambiguation is not done.

1.2 Motivation

Suppose a local user wants to get list of unique documents written on particular event for instance the Ethiopia's participation on the 14th World Indoor Game. The following two Amharic documents describe the same event written but written differently.

Document 1

ዘንድሮ በቱርክ በተደረገው የአለም ቤት ውስጥ ሻምፒዮና ኢትዮጵያ አመርቂ ውጤት አስመዝግባለች። በሻምፒዮናው 172 አገሮች የተሳተፉ ሲሆን 26ቱ ብቻ ሜዳሊያ ሠንጠረዥ ውስጥ ገብተዋል። ኢትዮጵያም አምስት ሜዳሊያዎችን በማግኘት በሶስተኝነት ውድድሩን አጠናቃለች። በውድድሩ ወጣቱ መሐመድ አማን በአንድ ደቂቃ 48.36 ሰከንድ አሸናፊ በመሆን ኢትዮጵያ ጎልታና ደምቃ ትታወቅበት ከነበረው ከረዥም ርቀት ባሻገር በመካከለኛ ርቀት ባለተስፋ መሆኗን አመላክቷል። መሐመድ አምና በዓለም ወጣቶች ሻምፒዮና የብር ሜዳሊያ በማግኘትም ለአገሪቱ የመጀመሪያዋ የ800 ሜትር ሜዳሊስት ለመሆን የቻለ ነው። የ3,000 ሜትር ሴቶች ሩጫ በታሪክ በተከታታይ ለአምስተኛ ጊዜ ያሸነፈች የመጀመርያዋ ሴት ለመባል ቆርጣ የነበረችው ኮከቧ መሠረት ደፋር ባልተጠበቀ ሁኔታ በኬንያዊቷ ሄለን ኦሳንዶ በአንድ ሰከንድ ተቀድማ ሁለተኛ ደረጃን አግኝታለች። በዚሁ ርቀት ገለቴ ቡርቃ በሦስተኛነት አጠናቃለች። የሴቶችን 1,500 ሜትር የኦሊምፒክ ድርብ ወርቅ ባለድሏ ጥሩነሽ ዲባባ እህት ገንዘቤ ዲባባ አሸንፊ ስትሆን በወንዶች 1,500 ሜትር መኩንን ገብረመድኅን በሦስተኛነት ነሐስ ሜዳሊያ አግኝቷል።

Document 2

ኢትዮጵያ በኢስታንቡል በተካሄደው 14ኛው የዓለም የቤት ውስጥ አትሌቲክስ ሻምፒዮና በተለያየ ርቀት ተሳትፊ ሁና አስደሳች የሆነ ውጤት በማግኘት ተመልሳለች። ገንዘቤ ዲባባ በአንድ ሺህ 500 ሜትር፣ መሐመድ አማን ደግሞ በ800 ሜትር ያስገኟቸው የወርቅ ሜዳሊያዎች የአገሪቱን ደረጃ ከፍ ለማድረግ አስችለዋል። በአጠቃላይ የሜዳሊያ ድምር 2 ወርቅ፣ 1 ብርና 2 ነሐስ በማግኘቷ ኢትዮጵያ ከአፍሪካ ቀዳሚውን ስፍራ ስትይዝ በዓለም አቀፍ ደረጃ ደግሞ ከአሜሪካና ከእንግሊዝ ቀጥላ የሦስተኛ ደረጃ የሚያስገኙላትን ውጤት አስመዝግባለች። ጎረቤት ሀገር ኬንያ ሁለት የወርቅ፣ አንድ የብርና አንድ የነሐስ ጨምሮ አራት ሜዳሊያዎችን በማግኘት የአራተኛ ደረጃ አግኝታለች።

Table 1-1: Sentences extracted from the sample documents

	Document 1	Document 2
S_1	ዘንድሮበቱርክበተደረገውየአለምቤትውስጥሻምፒዮናኢት ዮጵያአመርቂውጤትአስመዝግባለች።	ኢትዮጵያበኢስታንቡልበተካሄደው 14ኛውየዓለምየቤትውስጥአትሌቲክስሻምፒዮናበተለያየር ቀትተሳትፊሁናአስደሳችየሆነውጤትበማግኘትተመልሳለች ።

In order to identify the list of unique documents written on particular event, the use of traditional word occurrence based similarity is not enough. For instance, as shown in Table 1-1, S_1 in Document 1 and Document 2 are very similar as both documents uses different words/terms referring to the same fact (for example "አመርቂ" and "አስደሳች"). In addition, the two documents refers to concepts refers related issue – these concepts are normally related with semantic relation (i.e. Hyponym), such as the words "በቱርክ" in Document 1 and "በኢስታንቡል" in Document 2 those are not similar unless we apply concept based similarity which considers semantic relations between words.

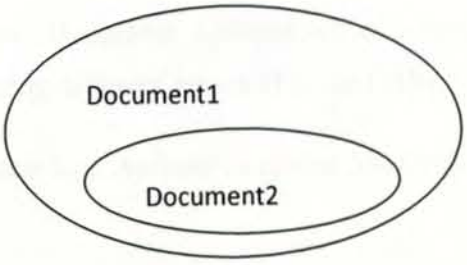


Figure 1-1: Relationship between Document1 and Document 2

The other thing we considered is use of asymmetric similarity measure instead of symmetric. In symmetric if a document A is similar with a document B, B is also similar with A. However, it is not sometimes possible to say this because these two documents may not be equally similar. For example, when we look at the above two Documents (Document1 and Document2), Document1 contains all sentences of Document2 and some other additional sentences as well. The relationship between these documents is illustrated in Figure 1-1.

As shown in Figure 1-1, Document2 is included in Document1, thus Document2 is identical to Document 1 but not vice versa.

This scenario shows the need to analyze conceptual contents of the documents in computing degree of similarity between documents.

1.3 Statement of the problem

Measuring text document similarity is a challenging task that requires general and domain-specific knowledge and deep analysis of documents. Learning from text written in natural language is one of great challenges in AI and machine learning. In particular, there is difficulty on addressing lexical and semantic levels in the text. Here, the problems come from polysemous (i.e. a word may have multiple senses and multiple type of usage in different context), synonyms (i.e. different word may have similar meaning) and some other semantic relationship that determine the meaning of the document. These difficulties are also shown in measuring similarity of documents as it requires addressing lexical and semantic information of documents.

Different researches have been done on processing Amharic documents. But as discussed in Section1.1, a correct sense of a word is not identified. For instance, in the research "Concept-Based Automatic Amharic Document Categorization", categorization of documents was performed without considering different senses of a word [**19**].

This study proposes a Concept-base Amharic Document Similarity (CADS) measure that handles the issue of WSD.

1.4 Objectives of the Study

1.4.1 General Objective

The general objective of this research is to develop a document similarity metric that consider semantic information embedded in the documents.

1.4.2 Specific Objectives

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

- ✓ Study different approaches that are used in developing Concept-base Document Similarity.
- ✓ Extend or adopt WordNet for Amharic thesaurus (AmhWordNet).
- ✓ Propose an approach/ algorithm that measure semantic similarity between documents.

- ✓ Develop a prototype for Concept Base Amharic Document Similarity based on the proposed algorithms.
- ✓ Prepare test corpus
- ✓ Collect data from Amharic dictionary to implement AmhWordNet.
- ✓ Evaluate the performance of the developed approach using standard measures such as precision and recall on test corpus.

1.5 Scope and Limitation of the Study

Scope

The developed system takes input of two Amharic text documents which have a domain of sport and measures their similarity semantically. In this research work, an Amharic WordNet is developed which is limited to words/terms that are related to sport domain.

Limitation

Other type of documents such as images, figures and tables are not considered in the similarity measure.

1.6 Methodology

For the accomplishment of the research, the following methodologies will be applied.

1.6.1 Literature Review

Detailed literature review in the area of computing document similarity will be conducted for deeper understanding of concept based document similarity measures. Particular focus will be given in reviewing literature for the techniques of document similarity measures, strategies to extend WordNet, writing system of Amharic languages and existing researches on concept based text similarity measure.

1.6.2 Data Collection

Data to populate the WordNet is required. This data is taken from Amharic dictionary [**20**]. In addition, Amharic sport domain corpus gathered from Walta Information Center (WIC) is used in implementing Pointwise Mutual Information (PMI) for evaluation purpose. PMI need collection of documents for measuring similarity as it is a statistical method. .40 different sports news are also collected to test the performance of the system.

1.6.3 Design and Implementation

An algorithm is designed for the purpose of implementing our system based on the identified research problems.

1.6.4 Tools and Techniques

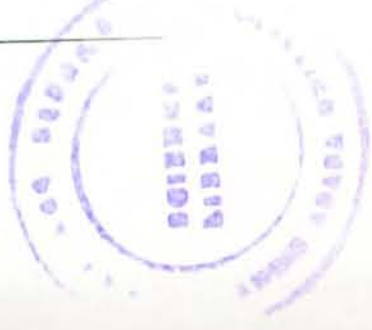
Different tools and techniques are used to achieve the goal of the research. The main parts of the system such as Word Sense Disambiguation (WSD), concept tree extraction, finding semantic relation between words and computing semantic similarity are developed with python. SQL server is used to design the WordNet.

1.6.5 Experimental Evaluation

The performance of developed system is evaluated using different evaluation metrics. The precision, recall and f-measure of the clusters of documents that are grouped using the similarity score of the documents measured by CADS, CADSWoWSD(CADS without WSD), Pointwise Mutual Information (PMI), Cosine and Jaccard similarity measures over human made clusters is calculated. 40 sports news are given to human and these news are grouped into 9 different clusters.

1.7 Application of the Result

Generally, document similarity metrics can be applied in the area of information retrieval, conceptual document clustering, and document recommendation systems. For example when we look for a page using a search engine, we are measuring similarity between a query document and a corpus document; when we search for a string in a text, we are measuring similarity between a search string and sub-string of the text; and when we summarize a document, we



produce a semantically similar document to the original document. It also can be used for classification, summarization, and for automatic evaluation of machine translation [**21**,**22**].

Therefore, this research has significance on the applications listed above. It can be used as an input for those applications.

1.8 Thesis Organization

The remaining part of this thesis is organized as follows:

Chapter Two: presents detailed discussion of literatures related to this work. It includes word and text similarity measures, text similarity approaches, similarity types such as symmetric and asymmetric, Amharic languages (its structure and difficulties), and NLPs like part of speech tagging and word sense disambiguation for similarity measures. This chapter also presents review of related works specific to semantic similarity in English and Chinese language.

Chapter Three: describes the Design and Implementation of the proposed CADS system. Algorithms like WSD, Concept tree extraction, similarity measures are proposed in order to implement the system.

Chapter Four: presents the experiments and results of the proposed system.

Chapter Five: presents a conclusion to the research work by discussing the main results that were obtained. Recommendations also made for future research related to this study.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

2.1 Overview

In this chapter, the literature reviews is conducted to provide information on different topics related to this research such as similarity measure, text similarity approaches, similarity type, Amharic language and its writing system and NLPs for similarity measures are presented. Related works also presents in this chapter.

Similarity can be computed between words, sentences or documents. Measuring of similarity between words is a base for measuring between sentences and documents.

2.2 Similarity Measure

2.2.1 Word Similarity Measures

Similarity between words can be calculated from using the syntactic information (i.e. letters) or semantic (meaning) of words. A method like Edit distance (also called Levenshtein distance) and Longest Common Subsequence (LCS) [**23**] compute similarity between words from the spelling of the words where as a lexical dictionary like WordNet used to compute similarity between words based on their meanings [**24**].

Semantic similarity between terms/concepts can be calculated based on distance, information content or feature.

Distance Based

The similarity measure behind distance based method is to find a length between a pair of concepts in a given thesaurus. Leacock and Chodorow [**25**] , Wu and Palmer [**26**] and path length [**27**] are similarity measures that are used distance based method.

The path length is a baseline that is equal to the inverse of the shortest path between two concepts. The Leacock and Chodorow method finds the shortest path between two concepts and

scale the path length value by the maximum path length found in the is-a hierarchy in which they occur.

The Wu and Palmer method measures how similar two concepts C1 and C2 are based on the depth of the two concepts in the taxonomy (i.e. the depth of a concept is simply its distance to the root node) and that of their Least Common Subsumer of the concepts.

$$Sim_{wp}(C_1, C_2) = \frac{2 \times N}{N_1 + N_2} \qquad 2\text{-}1$$

Where,

R -the root concept in the concept tree

N -Len(R, C)

C - LCS(C1, C2)

N1– Len(R, C1)

N2 – Len (R, C2.)

Information Content Based

Information Content Based method is proposed by Resnik in 1995 to solve the drawback of distance based methods that relies on the idea that links in the taxonomy represent uniform distance [**27**]. The method used information content to evaluate semantic similarity by associating appearance probabilities to each concept in the taxonomy, computed from their occurrences in a given corpus.

Feature-Based

Feature-Based methods measures similarity between concepts as a function of their properties (i.e. their glosses in the WordNet) or based on their relationships to other relationships to other similar terms in the taxonomy. The method relies on the matching between synsets and a concept's glosses extracted from WordNet. In feature-based methods, concepts are similar is they share synsets and glosses [**28**].

Pointwise Mutual Information (PMI)

PMI is unsupervised statistical approach which is used to find semantic relatedness between words [29]. In PMI, calculating semantic similarity between words is done by considering the words marginal frequencies and their co-occurrence frequency in a corpus. This means two concepts are more likely to co-occur in a common, shared context and less likely in an unshared one. It is mathematically expressed as Equation 2-2

$$PMI(x,y) = \log \frac{P(x,y)}{P(x).P(y)} \qquad 2\text{-}2$$

Even though, PMI has been used by different researchers to calculate similarity between words [30,29], it has drawback. The high score of PMI sometimes might not necessary indicate the two objects are related.

2.2.2 Text Similarity Measures

The similarity between texts can be computed from similarity between words. If words in two texts are similar, the two texts are more possibly similar. But it does not means that word similarity measure replace text similarity measure since there are words that have multiple meanings and need to identify the exact meaning the words have in that context.

Different measures of similarity between texts are proposed in different researches. Cosine, Jaccard, and Matching average similarity measures are some of them.

Matching Average

Matching Average is used to compute the similarity between two texts. It is used in [31] to calculate similarity between sentences for Recognizing Textual Entailment System. It is defined as Equation 2-3.

$$MatchingAverage = 2 \times \frac{Matching(s1, s2)}{length(s1) + length(s2)} \qquad 2\text{-}3$$

Cosine Similarity

The cosine similarity is one of the simplest ways of computing similarity between documents as the cosine of the angle between their corresponding word vectors. One of its main advantages is that it is domain and model free [**32**].

$$CosSim(A,B) = \frac{\mathbf{A}.\mathbf{B}}{|\mathbf{A}|.|\mathbf{B}|} \qquad 2\text{-}4$$

When cosine similarity is used to calculate similarity between texts, each text first represented by a number of words in sequence and similarity computing is done based on cosine similarity formula. The formula is shown in Equation 2-4

Jaccard Similarity Coefficient

Jaccard similarity coefficient is asymmetric similarity measure that is defined as the size of the intersection divided by the size of the union of the sample set.

Equation 2-5 shows mathematical expression of Jaccard coefficient.

$$JaccSim(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad 2\text{-}5$$

When using Jaccard similarity metric to measure similarity between texts, each text first represented by a number of words in sequence like cosine similarity then distinct words are extracted from both texts and finally computing the Jaccard similarity using Equation 2-5 is done.

2.3 Text Similarity Approaches

Approaches such as statistical, knowledge based and hybrid are available to measure similarity between texts.

2.3.1 Statistical Approaches

This approach requires a collection of documents (corpus) to compute document similarity. In [33] statistical approaches based on word set, word vector, edit distance, word order and word distance had implemented to measure sentences similarity.

In the next sub sections, Vector Space Model (VSM) and Latent Semantic Analysis (LSA) which are among the methods used for implementing statistical approaches are discussed.

Vector Space Model (VSM)

VSM is a word co-occurrence method in which documents are represented by vector of keywords which are extracted from documents with associated weights that represent the importance of keywords. This method calculates similarity according to the number of occurrences of terms in the specified document or text. The term weighting for the vector space model has entirely been based on single term statistics. One of the most widely-used term weighting heuristics is called the term-frequency inverse document-frequency (tf-idf) introduced by Salton and McGill in 1983 [3]. The sense of tf-idf is that the more the term occurs in the document the more weight it has. VSM use different similarity measures such as cosine similarity, Jaccard similarity, dice similarity and overlap similarity measures.

In the model semantics and word order are ignored and text is treated as a bag-of-words (BOW). The VSM relies on the assumption that more similar documents have more words in common but this is not always the case. It ignore the potential semantic relations between words such as synonyms (terms with different spelling having a same sense like the terms big and large), polysemy (terms having different sense with a same spelling such as apple as a fruit and apple as a company), etc. Due to such things, similar phrases may be treated as dissimilar. For example, phrases 'I have a cat' and 'I own an animal' are treated completely as dissimilar because they share no words.

Latent Semantic Analysis (LSA)

LSA is the extended form of VSM that analyze relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms[34]. This approach addresses partially the problem of synonym and polysemy that occur in VSM. In

addition, in VSM, the document matrix is usually of a high dimension and sparse, since every word does not appear in each document. High dimensional and sparse matrices are susceptible to noise and have difficulty in capturing the underlying semantic structure. Moreover, the storage and processing of such data place great demands on computing resources. The LSA uses a dimensionality reduction procedure such as Singular Value Decomposition (SVD) in order to reduce the sparse term-document matrix, which can be high-dimensional, to a lower-order matrix approximating the original. When measuring semantic similarity between documents using LSA, each document is represented as a vector in reduced-dimension space and similarity is then measured by using cosine of the angle between the documents corresponding row vectors.

Applicability of LSA is restricted due to SVD (i.e., SVD computation is demanding both in terms of processor time and in memory requirements).

2.3.2 Knowledge Based Approach

This approach addresses the problem of synonym and polysemy those are important for semantic document similarity measuring. Unlike the approaches that are discussed previously which relay on statistical distribution properties of terms, knowledge based approach uses background knowledge to find out the relatedness of terms. It uses resources such as WordNet [**35**] which is particularly created to reflect the relationship between terms and Wikipedia [**36**] which is knowledge repository on the Web.

WordNet

WordNet is a lexical database that is first created at Princeton University to use it as lexical knowledge of native English speaker. In the WordNet different information like English nouns, verbs, adjective and adverbs are organized into set of synonyms which are called synsets. A given synset consists of list of synonymous word forms and semantic pointers. A word form can be single word or two or more words connected by underscore and word definition. A semantic pointer tells the relationship between words within the synset and with other synset.[**35**]

WordNet has been used as input to measure semantic similarity in different researches. One of research that used WordNet for measuring semantic similarity between texts is [**37**]. In this paper

the semantic and syntactic information were taken considered to measure text semantic similarity.

Wikipedia

Wikipedia is another resource to measure semantic relatedness between words. According to different researches proof, it is a better knowledge base than WordNet. Researchers have used Wikipedia to compute semantic relatedness by using its hyperlink structure and category hierarchy or textual content. For instance, Strube and Ponzetto were the first to compute measures of semantic relatedness using Wikipedia [**38**]. They applied the WordNet techniques by modifying it to fit Wikipedia. Wikipedia Link-based Measure (WLM)[**39**] is introduced for extracting semantic relatedness measures from Wikipedia through Wikipedia's hyperlink structure.

2.3.3 Hybrid Approach

This approach is the combination of the above two approaches statistical and knowledge based. In [**10**] the researchers suggested this combined approach for measuring semantic similarity between texts. In the research the result of two corpus-based measures; PMIIR and LSA and six knowledge based measures; the Leacock & Chodorow, the Lesk, the Wu and Palmer, Lin, Resnik, and Jiang& Conrath of word semantic similarity are combined to show how hybrid approach can be used on measuring texts similarity.

The semantic similarity between two sentences is calculated in [**40**] using information from lexical database and corpus. The lexical database (WordNet) is used to calculate the semantic similarity and the corpus is used to provide information content. The researchers considered three similarity functions (string similarity, common word order similarity and semantic similarity).

Researchers in [**10**] used the hybrid approach that is the combination of corpus-based and knowledge based to measure text semantic similarity. In the research, two corpus-based measures and four knowledge based measures are combined to measure text semantic similarity. Combination of corpus-based word similarity and string similarity is presented in [**23**] to compute semantic similarity between sentence.

2.4 Similarity Types

The similarity type can be either symmetric or asymmetric based on directional similarity [**41**].

Symmetric Similarity

In symmetric similarity, the measure of two object A and B is equal to the similarity between B and A (i.e. sim(A,B) = sim(B,A)). Symmetric similarity only concern in the common features of objects and it ignore the distinct features.

Asymmetric Similarity

In asymmetric similarity the similarity between A and B equal to similarity between B and A if and only if object A is equal to object B (i.e. sim(A,B) = sim(B,A) iff A=B) . This similarity considers both common and distinct features of the two objects. Directional measure of similarity, which indicates the semantic similarity of a text segment A with respect to a text segment B is defined in the research [**10**] in order to measure text similarity. This asymmetric similarity provided the researchers with the flexibility they needed to handle application where the directional knowledge is useful.

2.5 Amharic Language

2.5.1 Overview of Amharic Language

Amharic, one of Semitic languages is spoken in many parts of Ethiopia. It is the second most spoken Semitic language in the world next to Arabic [**42**]. It is the official working language of the Federal Democratic Republic of Ethiopia.

The language is written in the unique and ancient Ethiopic script which is called fidel, inherited from Geez language which is currently used only in Ethiopian Orthodox Tewahedo Church as worshiping language. Amharic script has 33 core characters and 32 of them are consonants having seven orders to show the seven vowels. Out of the seven derivatives, six of them are CV (Consonant vowel) combinations while the sixth is the consonant itself. Other symbols representing labialization, numerals, and punctuation marks are also available.

2.5.2 Grammatical Structure

The Amharic grammatical structure is organized into five basic part of speech such as noun (ስም), adjective (ቅፅል), adverb (ተውሳከግስ), verb (ግስ) and preposition (መስተዋድድ)[**43**]. The word order in Amharic clauses is generally Subject Object Verb (SOV). In Amharic grammar, verbs agree with their subjects in number gender and person and objects precede verbs within the verb phrase.

2.5.3 Punctuations

Different punctuation marks are available in Amharic language writing system. The punctuation : "ሁለት ነጥብ" is used to separate words but this is not practically used in type written texts instead blank spaces are used. The "አራት ነጥብ" symbol, ።, is used to separate sentence, and the symbols ፣ and ፤ which are called as "ነጠላ ሰረዝ" and "ድርብ ሰረዝ" respectively are equivalent with comma and semicolon in English. In addition, punctuations like ?,! , " , ", ', /,and \, which are borrowed from foreign languages also used in Amharic language.

2.5.4 Difficulties of Amharic Language

In Amharic language there are issues that make the language difficult to use it in NLPs. One of the issues is a word can be written in more than one ways because of availability of words with similar sound but different alphabets (i.e., The alphabets (the so called fidels) "ሀ" "ሐ" and "ኀ","ሰ" and "ሠ","አ" and "ዐ","ጸ" and "ፀ" can be used interchangeably so that any two words formed from these interchangeable alphabets are considered as similar. For example the words "Hager" and "Hager" are similar words with different symbols. The other issue is some words have a short form which are written using the symbols '/' or '.' For instance a word "ዶክተር" has a short form "ዶ.ር" or "ዶ/ር". These issues should be handled when using the language in NLP tasks (in similarity measure and in other NLP applications).

2.6 NLPs for Similarity Measures

Measuring similarity between words, texts or documents require the knowledge of how we understand the meaning of words and sentences. NLP gives an answer for this by providing different applications such as Part of Speech (POS) tagging and WSD. In our research we used these NLP applications.

Part of Speech Tagging

Words in a language are grouped into word classes commonly known as POS in which words in similar word classes have similar syntactic behavior [**44**]. POS tagging is a process of assigning a POS to a given word. Approaches such as statistical, rule based and hybrid are available to automate POS tagging.

POS tagging make similarity measure easy. This is because, when comparing two words from tagged documents, if the words have different POS we can directly say they are dissimilar without doing further analysis but if the words are untagged it needs extra processing. Moreover, POS tagging helps to implement WSD. During identifying sense of polysemous word from a document, if we know the POS of the word in the document, WSD can focus on only the class that the word has rather than worrying about other classes.

Word Sense Disambiguation

There are words in natural languages which are polysemous, (i.e. words with multiple meanings or senses). The process of automatically determining the sense of a polysemous word is called word sense disambiguation [**45**]. Disambiguation is done by looking at the context of the words.

The three approach of WSD are corpus based, knowledge based and hybrid approaches. The corpus based approach relays on the availability of text data (corpus). Knowledge based approaches use lexical database such as WordNet. The Lesk algorithm [**46**] is introduced by Michael E. Lesk that used knowledge based as a resource and disambiguates polysemous words in a sentence context. To disambiguate a word using Lesk algorithm, the gloss of each of its senses is compared to the glosses of every other word in a phrase. Hybrid approach is the combination of the corpus and knowledge based approaches. WSD is important in semantic similarity measures since the measures rely on the meaning of words.

2.7 Related Works

There are researches that are carried out on semantic texts similarity measure i.e. based on what words on the document means in English and other languages. Semantic similarity measure was proposed to solve problems of handling semantic nature of texts during similarity measure using traditional method (i.e. word occurrence method).

Sentence Similarity Based on Semantic Nets and Corpus Statistics

In this research, a method is presented to perform semantic similarity of two sentences. The proposed method computed text similarity from semantic and syntactic information of each text. The semantic information is derived from corpus whereas the syntactic information is formed from Lexical database [**40**].

Given sentence s1 and sentence s2, first the method dynamically formed a join word set using all the distinct pair of s1 and s2. Then raw semantic vector and word vectors are produced for s1 and s2 with the help of lexical database. The significant of a word is weighted from corpus using information content since each words in a sentence contributes differently to the meaning of the whole sentence. After computing semantic similarity from the two semantic vectors and an order similarity from the two order vectors, the sentence similarity is produced combination of semantic and order similarities.

Word Sense Disambiguation-based Sentence Similarity

None of the above researches considered the actual meaning of the sentences to compute similarity rather they relied on the nearest meanings of words. This research was conducted to explore how similarity is computed based on the actual meaning [**48**]. To do so, a method which is called Word Sense Disambiguation-Semantic Text Similarity (WSD-STS) proposed. In WSD-STS, first transforming an existing corpus-based measure (i.e. STS) into knowledge-based measures is done and then word sense disambiguation is integrated.

Given sentence s1 and sentence s2, first the string similarity and semantic similarity of s1 and s2 is computed then the overall sentence similarity is measured based on those similarities.

2.7.1.2 Semantic Text Similarity for Chinese Language

The Research of Chinese Semantic Similarity Calculation Introduced Punctuations

In this research, researchers work on the semantic similarity calculation between sentences by introducing punctuations in Chinese language. In the research, the first thing which was done is considering the phrases in each sentences and if the phrases are the same, punctuation is

introduced to identify whether the sentence similarity is 1 else semantic similarity was calculated with the help of WordNet called HowNet[**49**].

A method of Phrased Integrated Semantic Similarity Computation

In this research a method of phased integrated semantic similarity computation is proposed to compute text similarity [**50**]. In the research, steps are performed during computation. First segmenting text into paragraphs, paragraphs into sentences, sentences into words is done. Then calculating the similarity of words, sentences, paragraphs to obtain the text similarity is performed. Finally computing text similarity is done.

Word similarity is computed using HowNet knowledge structure. In sentence similarity calculation first each sentence is represented as a vector then the average of the two vectors is taken. The overall text similarity is computed based on the result of sentence similarity.

CHAPTER THREE

DESIGN AND IMPLEMENTATION OF CADS

3.1 Overview

In this Chapter, our approach, Concept-based Amharic Document Similarity (CADS) model, is introduced. The model consists of three major components – Text Pre-processor, Amharic WordNet (AmhWordNet), and Concept-based Similarity Measure (CSM) as shown in Figure 3-1. The aim of the text pre-processor is to prepare the input documents for CSM. Tokenization, Normalization and Stemming are the major activities conducted in this component. Once this part is completed, the preprocessed documents are ready to be used.

The CSM component is responsible to compute the similarity of documents. This component is composed of Concept Tree Extraction (CTE), Word Sense Disambiguation (WSD), and Similarity Measure modules. Capturing polysemy, synonymy and other semantic relations have main contribution in order to meet the objective of CADS. The issues of polysemy, synonymy and semantic relations are captured by CSM modules. The CSM component is integrated with AmhWordNet. The main task of Concept Tree Extraction (CTE) module is to extract concept tree that shows hierarchy between concepts of AmhWordNet. WSD module identifies polysemous words from documents and associates the most probable sense to it. The Similarity Measure module calculates the overall similarity between documents using similarity between its constituting components - word and sentences. The module is an integration of different sub modules such as Semantic Word Similarity (SWS) module, Sentence Similarity (SS) module and Document Similarity (DS) module. SWS module is responsible to find out the semantic similarity between words of documents. SS is responsible to measure similarity between sentences by using the result of SWS and the responsibility of DS is to find out the overall similarity score between documents.

The Amharic WordNet contains Amharic words along with its different meanings - concepts and semantic relations within concepts. This component helps to implement the modules of CSM.

The general architecture of the system is shown in Figure 3-1.Detail description of each component of the system is presented in the next sub-sections.

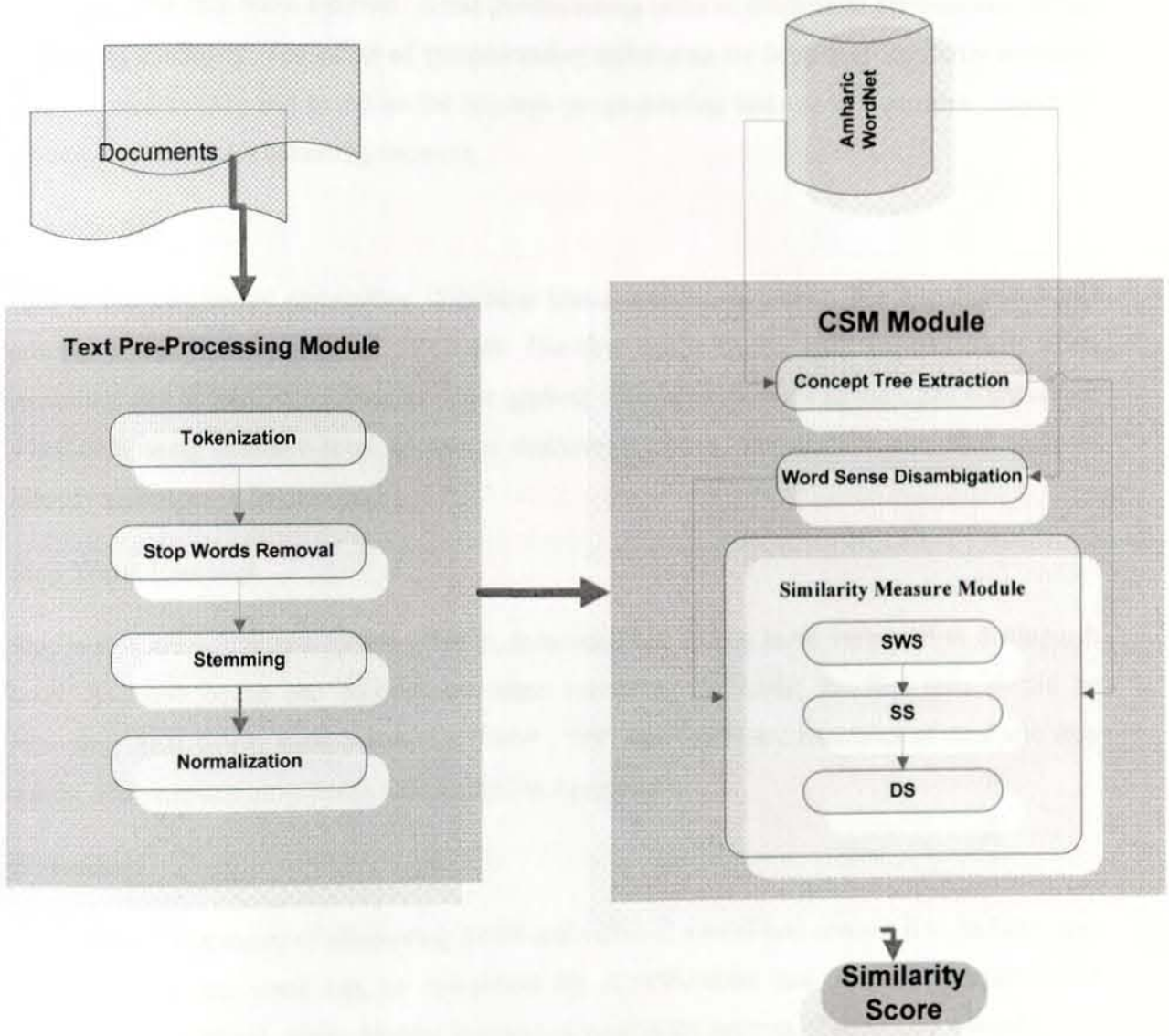


Figure 3-1: General Architecture of CADS

3.2 Text Preprocessing

The Text Preprocessing component of CADS is responsible to prepare the input documents for further processing by applying specific operations such as tokenization, normalization, stemming, and stop word removal. These preprocessing tasks in documents are common before computing similarity. The effect of pre-processing techniques on document similarity accuracy was assessed in [**51**] and based on the research pre-processing tasks have a positive impact on accuracy of document similarity measure.

Tokenization

This is the process of segmenting document into sentences or words. We implemented this process in the two components of CADS. The first one is in the Text Pre-processor; when stemming and normalization processes are applied. The second one is in the CSM component; when computing sentence level document similarity measure, tokenization is applied so as to identify sentences in documents.

Stop Word Removal

Stop words are words that appear often in documents but do not serve very well to distinguish texts. As these words are insignificant when measuring similarity; the first step should be removing these words from documents. "በጣም , "ግን" and "ሲሆን" are examples of Amharic stop words. The Amharic stop words are attached as Appendix C.

Stemming

Stemming is the process of eliminating prefix and suffix of a word and convert it to its base form to ensure that the word can be recognized by AmhWordNet and perform morphological matching using related words. Manual approach is used in the process of stemming a word.

Normalization

Normalization process is applied in order to handle issues which are discussed in Section 2.5.4. The issues are, in Amharic language a word can be written in more than one ways because of availability of words with similar sound but different alphabets and some words have a short form which is written using the symbols.

3.3 The Amharic WordNet

According to [52], using a WordNet for Amharic language has a significant impact on performance of tools such as search engine, automatic text categorization and Amharic word sense disambiguation. Likewise, applying WordNet on similarity measure is useful because using it, capturing WSD, synonym and semantic relationship between words are possible and these leads similarity measure to have better performance. Therefore, in this research we have adapted the structure of the English WordNet which is discussed in Section 2.3.2 and constructed Amharic WordNet in a way that it can be used as an input to build the CADS system.

3.3.1 The Structure of Amharic WordNet

Amharic WordNet (AmhWordNet) is a lexical database where each word is represented by its associated concepts. This is usable to remove ambiguity in cases where a single word has multiple meanings. For example the word "ዋና" has multiple meanings such as "በጉልህ የሚታይ አስፈላጊ የሆነ ዋና ጉዳይ", "መነሻ መነገጃ መንቀሳቀሻ ገንዘብ" and "በእጅ በእግር እየተቀዘፈ በውሃ በባህር ላይ የሚደረግ ስፖርት" and to handle such cases the word is associated to different concepts each representing separate senses.

Concepts[1] are connected to each other through semantic relations that are defined in the WordNet such as Hypernym, Hyponym, Meronym and Holonym

Synonym Relation:

The synonym relation is used to organize words with similar meanings under the same concept. Those words are referred to as synsets.

For example, in AmhWordNet, the word "ሀሴት" (WordId: w36) and "ደስታ" (WordId: w37) are under the same concept (ConceptId: c32) which is defined as "ፈንጠዚያ የሞላበት ሀዘን ትካዜ የሌለበት ሁኔታ".

[1]A concept is refers to set of words having similar meaning represented in the form of glosses or it can be collection of words defining similar sense

Hyponym (is a kind of) Relation

If a concept is connected to another concept through the *is-a-kind-of* relation then the relation is Hyponym. Hyponym relation captures subset taxonomy. A concept C_1 that contain synset $\{a_1,..., a_n\}$ is Hyponym of the concept C_2 that contain synset$\{b_1,b_2,...,b_n\}$ if every C_1 is a kind of C_2.

Hypernym is the reverse of Hyponym (i.e. Hypernym relation captures super set taxonomy). A concept C_1 that contain synset $\{a_1,..., a_n\}$ is Hypernym of the concept C_2 that contain synset$\{b_1,b_2,...,b_n\}$, if every C_2 is a kind of C_1. For example the concept containing {"ሩጫ"} is Hyponym of {"አትሌቲክስ"} and {"አ ት ሌ ቲ ክ ስ "} is Hypernym of {"ሩ ጫ"}.

Meronym (part-of) Relation

If a concept connected to another concept through the *part-of* relation then the relation is Meronym. A Concept C_1 represented by the synset $\{a_1,a_2,...,a_n\}$ is said to be a Meronym of a concept C_2 represented by synset$\{b_1,b_2,...,b_n\}$ if C_1 is a part of C_2.

Holonym is a reverse of Meronym relation. A Concept C_1 represented by the synset $\{a_1,a_2,...,a_n\}$ is said to be a Holonym of a concept C_2 represented by synset$\{b_1,b_2,...,b_n\}$ if C_2 is a part of C_1 . For example the concept C_1 {"ኢትዮጵያ"} is Meronym of the conceptC_2 {"አፍሪካ"} and {"አፍሪካ"} is Holonym of {"ኢትዮጵያ"}.

Figure 3-2 shows the semantic relations that are discussed above between words.

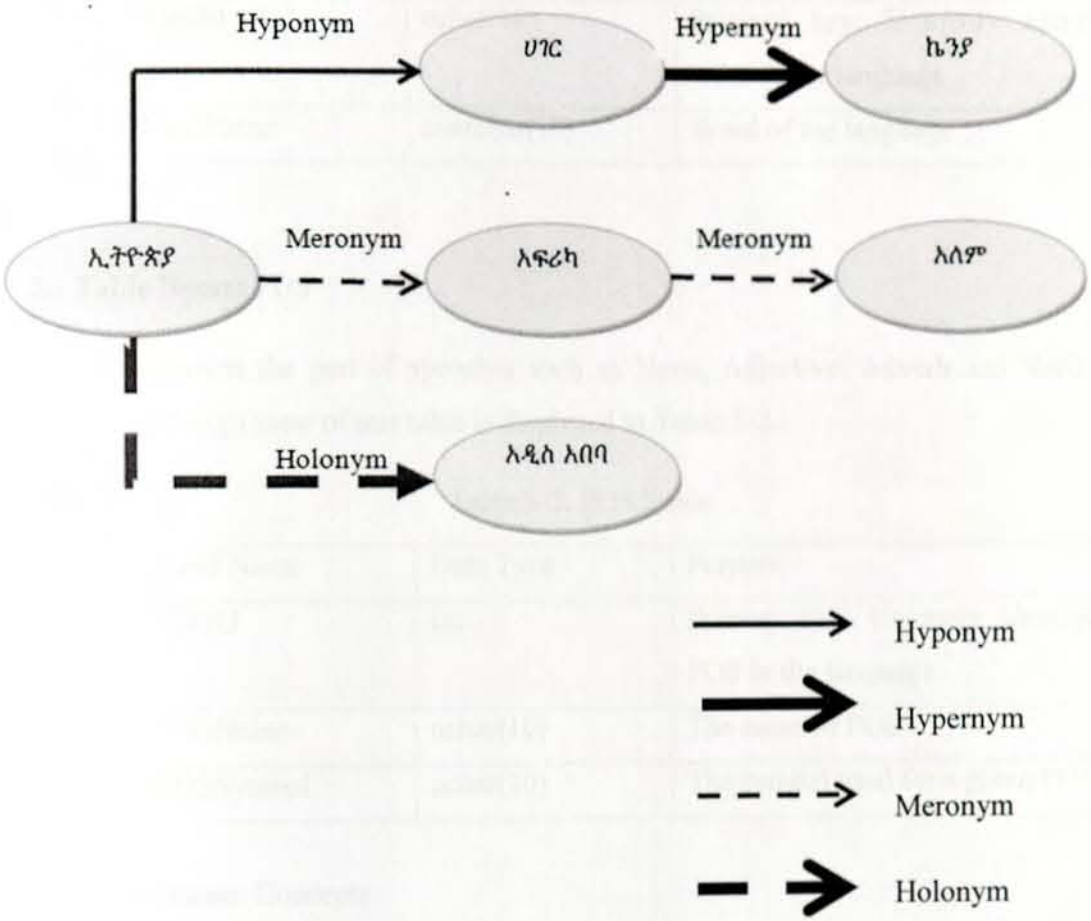


Figure 3-2: Example of Semantic Relations between concepts

3.3.2 Database Design of Amharic WordNet

The AmhWordNet database has five basic tables - Words, POS, Concepts, Word_Concept and SemanticRelations. The detail of each tables are shown presented as follows.

1. **Table Name:** Words

The Table "Words" is used to maintain unique words of Amharic language. As shown in Table 3-1, this table contains two fields- WordId and WordName.

Table 3-1: Word Table

Sr. No.	Field Name	Data Type	Purpose
1	WordId	nchar(10)	Primary key: Uniquely identifies a word in the language
2	WordName	nvarchar(10)	Word of the language

2. **Table Name:** POS

This table maintains the part of speeches such as Noun, Adjective, Adverb and Verb of the language. The design view of this table is displayed in Table 3-2.

Table 3-2: POS Table

Sr. No.	Field Name	Data Type	Purpose
1	POSId	Int	Primary key: Uniquely identifies a POS in the language
2	POSName	nchar(10)	The name of POS
3	POSSymbol	nchar(10)	The symbol used for a given POS

3. **Table Name:** Concepts

The purpose of this table is to maintain concepts which are used to describe a sense of words. The design detail of this table is presented in Table 3-3.

Table 3-3: Concept Table

Sr. No.	Field Name	Data Type	Purpose
1	ConceptID	nchar(10)	Primary key: Uniquely identifies a concept of words in the language
2	Gloss	nvarchar(MAX)	It explains the concept with definition that elaborate all words with given meaning or sense

4. **Table Name:** Word_Concept

The table is used to uniquely identify the sense of a word through the field WCId. Table 3-4 shows the design of the "Word_Concept" table.

Table 3-4: Word_Concept Table

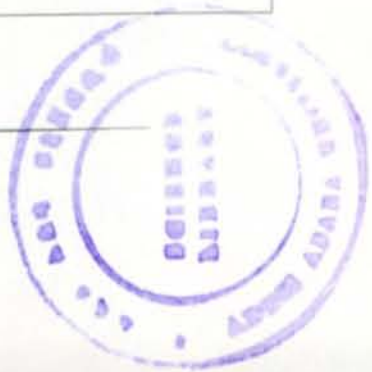
Sr. No.	Field Name	Data Type	Purpose
1	WCId	nchar(10)	Primary key: Uniquely identifies the sense of a word
2	WordId	nchar(10)	Foreign key from Word table.
3	ConceptId	nchar(10)	Foreign key from Concept table
3	POSId	Int	Foreign key from POS table.

5. **Table Name:** SemanticRelation

This table is used to maintain the semantic relation like Hyponyms, Hypernym, Holonym, and Meronym between pair of concepts. The fields which are used to build this table are shown in Table 3-5.

Table 3-5: SemanticRelation Table

Sr. No.	Field Name	Data Type	Purpose
1	RID	Int	Primary key: Uniquely identifies a relation between concepts
2	conceptID1	Int	Foreign key from Concept table indicate the first concept.
3	conceptID2	Int	Foreign key from Concept table indicate the second concept.
4	Relation	nchar(10)	It explains what two pair of concepts relation have



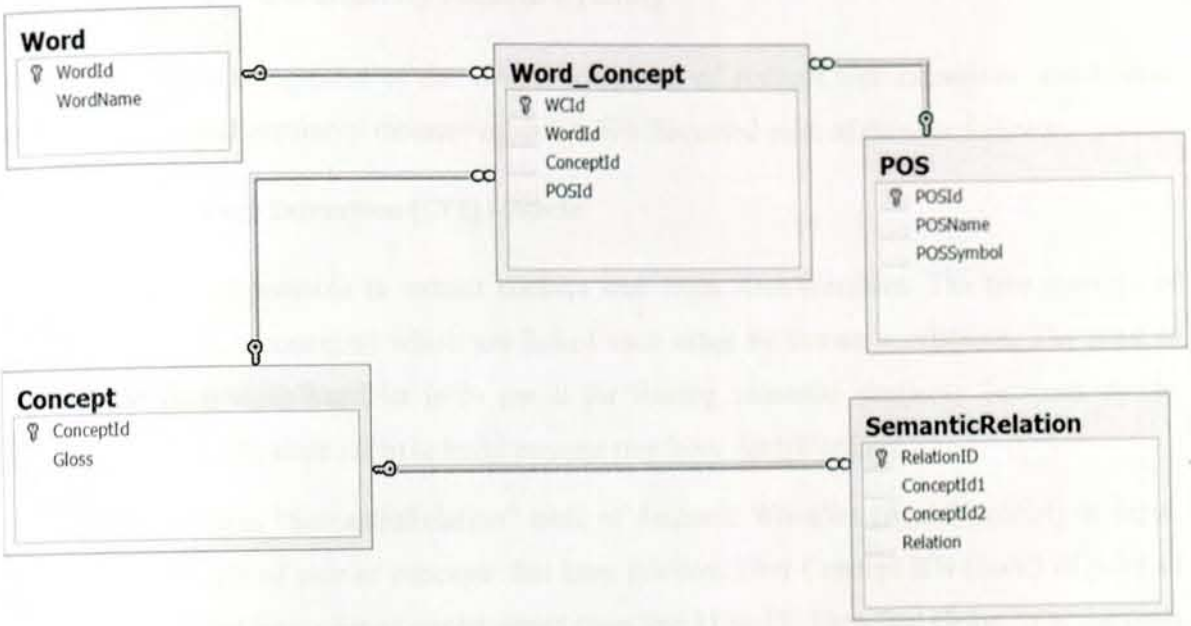


Figure 3-3: Database Schema of Amharic WordNet

Figure 3-3 shows the database schema of AmhWordNet. Different words in Amharic language are stored in the Word table identified by WordId field. Concepts are included in the Concept table and each concept is identified by ConceptId. POS table hold the name and symbol that represent the main POSs of Amharic language such as Noun, Adjective, Adverb and Verb. A word with a particular sense is identified with the help of Word_Concept table. SemanticRelation table hold pair of concepts along with the relation they have. The next scenario shows how these tables work together.

For example the word "ለጋ" has a WordId: w34 in the AmhWordNet. Since this word has two senses, the Concept table holds concept definition of those senses of the word under different ConceptId: c28with a Gloss"ኣስ በረጅም መምታት፤ ልጁ ኣስ መታ ወይም ለጋ" and c29 with a Gloss"ወጣት፤ ልጁ ወጣት ነው". Word_Concept table differentiate each sense of the word by combining WordId and ConceptID together and this combination is identified via WCId. This means the word "ለጋ" with a sense "ኣስ በረጅም መምታት" has a WCId: wc35 which is a combination of WordId: w34 and ConceptId: c28. And word "ለጋ" with a sense. "ወጣት" has a WCId: wc3 which is a combination of WordId: w34 and ConceptId: c29.

3.4 Concept-base Similarity Measure (CSM)

CSM is the main component of the CADS composed of concept tree extraction, word sense disambiguation and similarity measure modules. We discussed each of them as follows:

3.4.1 Concept Tree Extraction (CTE) Module

CTE module is responsible to extract concept tree from AmhWordNet. The tree consists of different nodes (i.e. concepts) which are linked each other by semantic relations. The need to extract tree from AmhWordNet is to use it for finding semantic similarity between words. Algorithm 3-1 shows steps taken to build concept tree from AmhWordNet.

The Algorithm takes "SemanticRelation" table of Amharic WordNet (AmhWordNet) as input. The table holds IDs of pair of concepts that have relation. First Concept IDs (node) of pairs of concepts are loaded into a list as implemented from line 11 to 17. Then first elements of the pairs as a key and the pairs as a value are loaded into a dictionary (from line 19-24). Next, each pairs are read from the list (line 26). Using a function "Get_Path", paths between nodes are found as implemented from line 31-40. Finally, those paths are written into a file (line 41).

Algorithm 3-1: Concept Tree Extraction Algorithm

```
1.      ALGORITHM(ConceptTreeExtractor)
2.      Input:
3.      WordNet: AmhWordNet
4.      Variables:
5.       L1,L2,L3,Linklist,Tree:List
6.      Path: Dictionary
7.      Output:
8.      Concept tree
9.      BEGIN:
10.           LOAD SemanticRelation Table TO Rows
11.           FOR EACH Row IN Rows
12.     APPEND Concept_ID1 IN L1
13.                   APPEND Concept_ID2 IN L2
14.                   APPEND L1 and L2 IN L3
15.                   APPEND L3 IN Linklist
16.                   L3 =[]
17.           END FOR
18.           Path ={}
19.           FOR EACH LL IN Linklist
20.     IF LL[0]NOT IN Path

21.                 Path[LL[0]]=[]
22.               END IF
23.               APEEND Tuple(LL) IN Path[LL[0]]
24.           END FOR
25.           Used=[]
26.           FOR EACH node IN Linklist
27.
28.           IF node NOT IN Used
29.                      Get_Path(node)
30.                   END IF
31.                   FUNCTION Get_Path(node)
32.                           IF node[1] IN Path AS Key
33.                              Next_node=Path[node[1]]
34.                              ADD Next_node IN Used
35.                              APPEND node[0] IN Tree
36.                              Get_Path(Next_node)
37.                           Else
38.                              APPEND node[0],node[1] IN Tree
39.                           END IF
40.                   END FUNCTION
41.                   WRITE Tree
42.                   Tree = []
43.           END FOR
44.     END
45.   END ALGORITHM(Concept Extraction)
```

Figure 3-4 shows sample tree that is extracted by CTE module. The tree hold nodes that have "is a kind of" relation. Later the result of CTE (i.e generated trees) is used during finding semantic similarity between words. The arrow in the figure represents hierarchy which is hyponym relation between concepts.

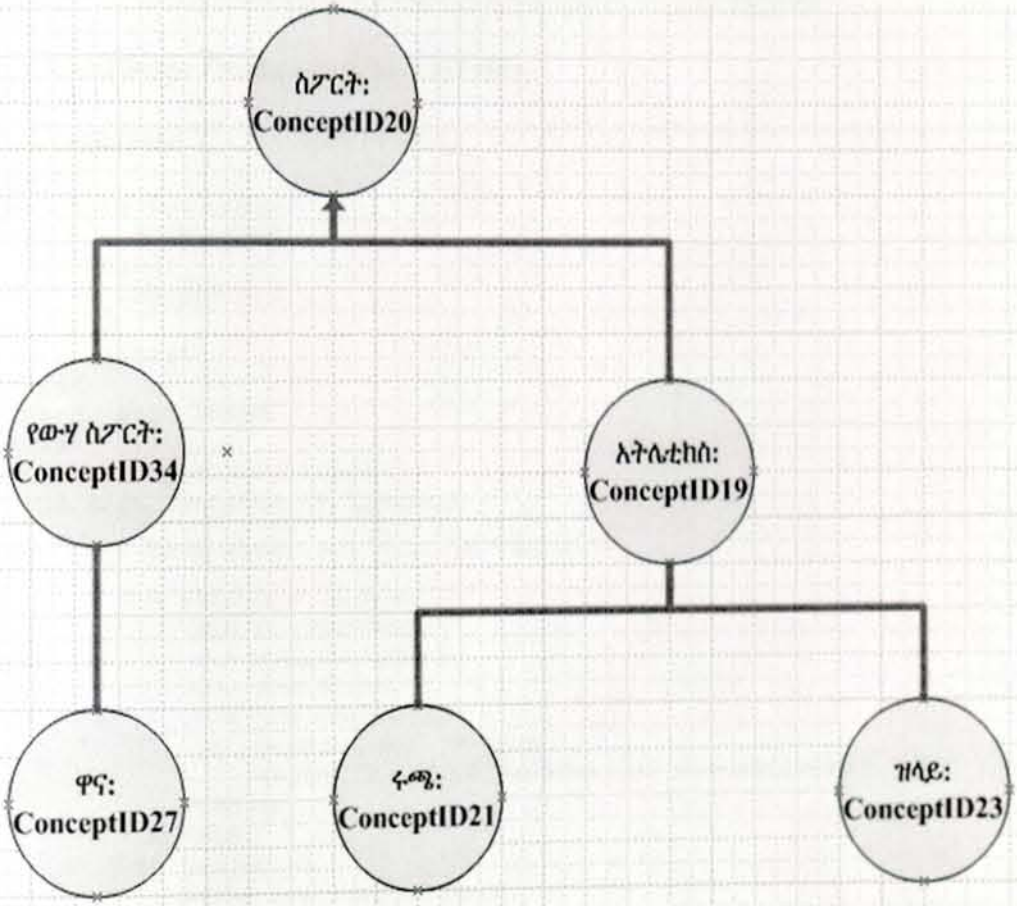


Figure 3-4: Sample Extracted Tree

3.4.2 Word Sense Disambiguation (WSD) Module

In this module identification of ambiguous words from documents and assigning the most suitable sense to them is done. Among the WSD approaches, we used knowledge base because it is possible to use our AmhWordNet for this purpose. We adopted gloss overlap approach of Lesk [46] to disambiguate words having several senses as presented in Algorithm 3-2.

Algorithm 3-2: Word Sense Disambiguation Algorithm

```
ALGORITHM(WSD)
  Input:
  Document: Doc1,Doc2
  WordNet: AmhWordNet
    Variables:
  max_overlap:int = 0
   i:int=0
  L1,L2,L3:List
  Output:
    Word, Best_sense
  BEGIN:
  READ Rows FROM Word Table
    FOR EACH Sentence IN Document
        FOR EACH Word, pos pair IN Sentence
          FOR EACH Row IN Rows
              W = Row.WordName
              P = Row.POSSymbol
              G = Row.Gloss
              C = Row.ConceptId
              IF  W==Word AND  P==pos
                  APPEND W,G,C IN Wo,glos,cpt respectively
              END IF
           END FOR
           IF len(Wo)>1
              WHILE n < len(sentence)
                    FOR EACH Row IN Rows
                        w1=row.WordName
                        p1=row.POSSymbol
                        g1=row.Gloss
                        c1=row.ConceptId
                        If w1==sentence[n] AND p1==senpos[n]
                           APPEND w1,g1,c1 IN wo1,glos1,cpt1respectively
                        END IF
                    END FOR
                    If len(wo1) > 0
                       WHILE C< len(glos)
                             g1=glos[c]
                             FOR g11 in glos1
                                 g11=g11.split(' ')
                                 inte=set(g1).intersection(set(g11))
                                 overlap=len(inte)
                                 If overlap > max_overlap:
```

```
43.                                     max_overlap = overlap
44.                                     best_sense = cpt[c]
45.                                 END IF
46.                             END FOR
47.                             c+=1
48.                         END WHILE
49.                     END IF
50.                     APPEND max_overlap IN maxsense
51.                     APPEND best_sense IN maxsense
52.                     APPEND maxsense IN maxbestsense
53.                     maxsense= []
54.                     wo1 = [], glos1=[], cpt1=[]
55.                 END WHILE
56.                 m =0
57.                 WHILE m < len(maxbestsense)
58.                     APPEND maxbestsense[m][0])IN  moverlap
59.                     m=m+1
60.                 END WHILE
61.                 mov=max(moverlap)
62.                 mm=moverlap.index(mov)
63.                 finalbestsense=maxbestsense[mm][1]
64.               WRITE Word,finalbestsense_
65.             END IF
66.           END FOR
67.       END FOR
68.     END
69.   END ALGORITHM(WSD)
```

Algorithm 3-2 is applied for both documents during similarity measure to disambiguate polysemous words. The following scenario shows how WSD is implemented.

The algorithm takes two documents as input and implements WSD on them to identify the senses of ambiguous words of the documents. After each document is tokenized into sentences, each word of a given sentence is identified whether the word is ambiguous or not. To do so, the first thing which is done by the algorithm is find the word from the WordNet by reading each word with its POS from the WordNet tables. Then if the word has more than one concept in the WordNet, the word is ambiguous (lines 14-23). The next thing which is done by algorithm is identifying the correct sense of the word which is implemented from line 24 to line 60 on the algorithm. To identify the sense, count the amount of words between the neighborhood words definition of each senses and the word definition of each senses from AmhWordNet. Then the best sense of the word that is to be chosen is the sense which has the biggest number of this count.

The following example illustrates Algorithm 3-2.

For a sentence S= "ስለሺ <N> ስሀን <N> በ1000 <NUMP> ሜትር <N > የብር <ADJ> ሜዳሊያ <N> አገኘ<V>" which is tagged manually, to identify the ambiguous words with its sense, assume the sentence is preprocessed, first we tokenize S into words then we get words= [ስለሺ, ስሀን, 1000, ሜትር, ብር, ሜዳሊያ, አገኘ]. Using AmhWordNet the ambiguous word in the sentence is "ብር" as it has more than one meaning ("ከአፈር ተነጥሮ የሚወጣ ከወርቅ ቀጥሎ ዝቅ ብሎ የሚገኝ የመአድን አይነት ፤ የተለያዩ ነገሮች እንደ ሜዳሊያ ሀብል የሚሰራበት", "ገንዘብ መገበያያ") with the same POS in the AmhWordNet. Then by applying Lesk algorithm (i.e. by taking the maximum overlap between gloss of concepts of words of the sentence and the gloss of each concept of the word "ብር"), the meaning of the word "ብር" in S is to refer the word "ሜዳሊያ".

3.4.3 Semantic Similarity Measure Module

This module is responsible to measure semantic similarity of two documents. We propose an algorithm Semantic Document Similarity Measures (SDSM) to perform the measurement. In the SDSM algorithm, documents are split into sentences and each sentence of one document is compared with the corresponding sentences of the other document. Finally, the document similarity is computed based on the similarity score of t sentences of the documents.

Steps for SDSM algorithm:

1. Split Document 1 into sentences $S_1, S_2, \ldots S_m$
2. Split Document 2 into sentences $S_1, S_2, \ldots S_n$
3. Compare each sentences of Document 1 with each sentences of Document 2. Comparing is done by computing sentences similarity using matching average metric based on Equation 2-3.
4. Take the maximum scores of sentences.
5. Calculate the overall document similarity score using Equation 3-1. The score is a value between 0 and 1.

To implement SDSM algorithm, modules such as Semantic Word Similarity (SWS), Sentence Similarity (SS) and Document Similarity are integrated.

Semantic Word Similarity (SWS) Module

SWS module is responsible to compute semantic similarity between words. We defined the score of similarity of words from 0 to 1. The similarity between two completely different words is 0 and two exactly similar words is 1 otherwise the score lay between 0 and 1.

Figure 3-5 shows how semantic similarity score between words is computed. As shown in the figure, first POS of the words are identified and if they are different no need to proceed to the next level since words with different POS are dissimilar and have 0 score. Once the POS of the words are identified and are the same, then the first task is to identify whether the words are the same or not. If they are the same, identify whether the word is ambiguous or not. If the word is unambiguous, the score will be 1. If the word is ambiguous, it has to be disambiguated using WSD module and then if they have the same sense the score will be 1 otherwise it will be 0. If the words are not the same, the next task will be finding their concept Id from AmhWordNet and if they are on the same concept they are similar else finding their semantic similarity using Wu and Palmer will be done through their concept Ids.

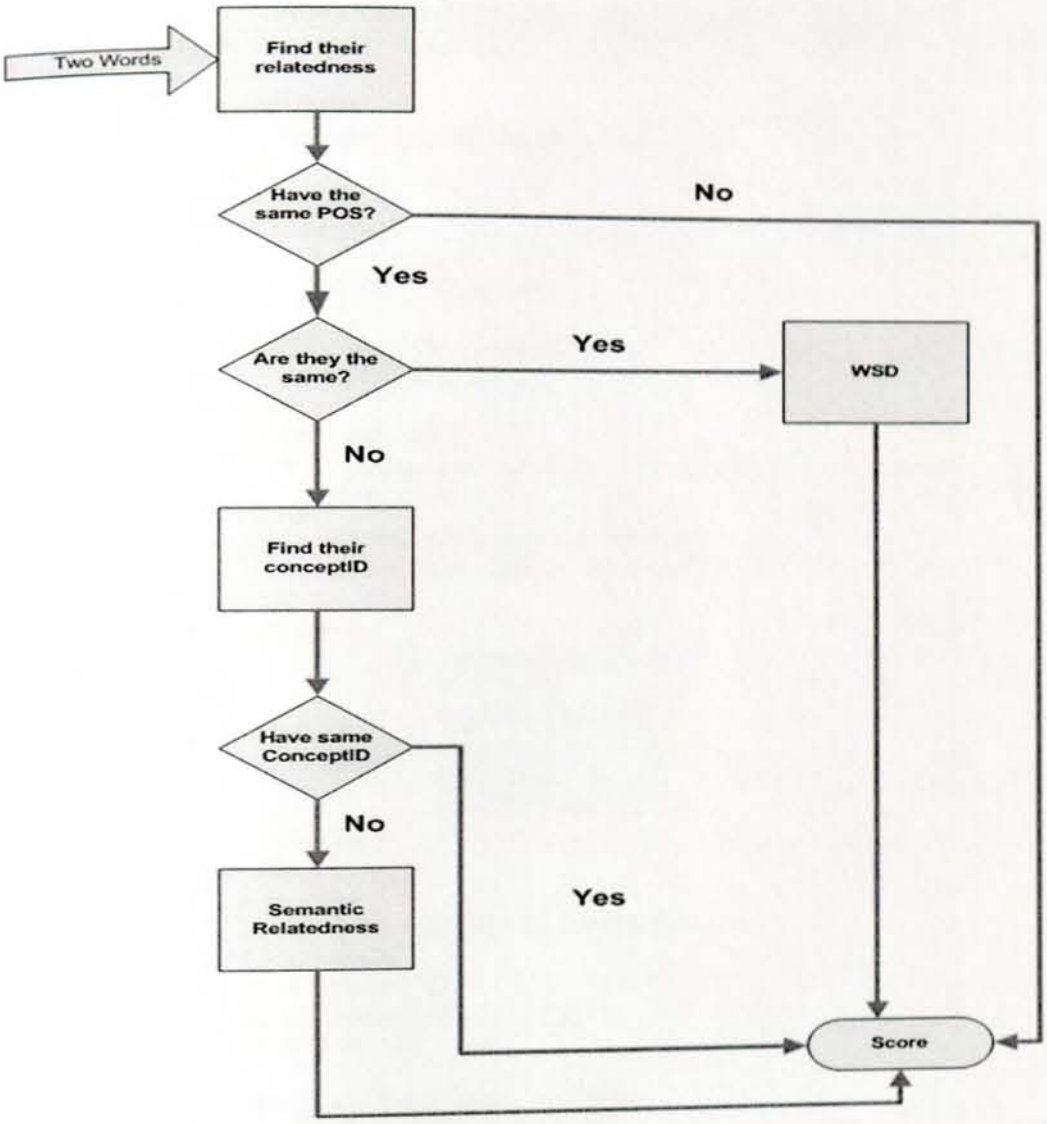


Figure 3-5: Flow of how semantic similarity between words is computed

The algorithm to calculate semantic similarity between words is presented in Algorithm 3-3.

Algorithm 3-3: Semantic Word Similarity Algorithm

```
1. ALGORITHM(Semantic Word Similarity)
2. Input:
3. Tree: ConceptTree
4. ConceptID1,ConceptID2, WCID1,WCID2:String
5.   Variables:
6. Concept, L:List
7. K:Boolean=False
8. Output:
9. Similarity score of two words
10 BEGIN:
11        FOR EACH Path IN  ConceptTree
12              APPEND Path IN L
13        END FOR
14        FOR EACH Path IN L
15             IF  ConceptID1 in Path and ConceptID2 in  Path
16                 K = True
17                 IndxCid1=Index(ConceptID1)+ 1
18                 IndxCid2= Index( ConceptID2)+ 1
19 BREAK
20             ELSE
21                      IF ConceptID1 IN Path
22                         P = True
23                         APPEND Path IN L1
24                      IF ConceptID2 IN Path
25                         Q = True
26                         APPEND Path IN L2
27             END IF
28        END FOR
29        IF K = True
30           IndxC = Min(IndxCid1,IndxCid2 )
31           N  =  IndxC
32           N1 =  IndxCid1
33           N2 =  IndxCid2
34           WP = 2*N/ N1 +  N2
35        ELSE
36            IF P == True AND Q == True
37               I=Intersection(L1,L2)
38               Indx = Index(I)
39               IF I NOT NULL
40 N  =  Indx + 1
41 N1 = N + 1
42 N2 = N + 1
43 WP = 2 * N/ (N1 + N2)
44               ELSE
45                  WP = 0
46               END IF
47            END IF
48       END IF
49       RETURN WP
50  END
51 END ALGORITHM(Semantic Word Similarity)
```

Algorithm 3-3 is used as input during calculating sentence similarity as illustrated in Algorithm 3-4. The algorithm takes two words together with their Concept ID from Algorithm 3-4 and its responsibility is to calculate the semantic similarity between words if those words have different Concept ID. The calculation is computed based on Wu and Palmer using Equation 2-1 from the concept tree which is already extracted using concept extraction module.

Sentence Similarity Module

After computing semantic similarity between words of the sentences, overall similarity measures of the sentences is computed using Matching Average (Equation 2-3). The following example illustrates how similarity between sentences is computed.

For example: S1 = ደሰታ <N> ዋና <N> ስፖርት <N> ይችላል <N>።

S2 = ሀይሉ <N> ስፖርት <N> ይወዳል <V>።

1. First each sentence is tokenized into words.
2. Compute similarity between each words of S1 and S2 as shown in the table

Table 3-6: Similarity Score between Words

S_1 \ S_2	ሀይሉ	ስፖርት	ይወዳል
ደሰታ	0	0	0
ዋና	0	0.5	0
ስፖርት	0	1	0
ይችላል	0	0	0

As shown in Table 3-6 the word "ዋና " and "ስፖርት" are semantically related. According to the concept hierarchy shown in Figure 3-4 and based on wu and palmer measurement (Equation 2-1) the score between these words is 0.5. i.e.

$$Sim_{wp}\left(\text{ዋና, ስፖርት} = \frac{2\times1}{(1+3)}\right) = 0.5.$$

3. Compute sentence similarity score using Equation 2-3 which is $\boldsymbol{MatchingAverage} = 2 \times \frac{\boldsymbol{Matching(s1,s2)}}{\boldsymbol{length(s1)+length(s2)}}$

Match (s_1, s_2) is the sum of maximum score of pair of words i.e. 0+0.5+1+0=1.5

length ($s1$)is 4 and length($s2$) is 3

Matching Average will be 2*1.5/7= 0.42.

The algorithm for semantic sentence similarity measure is presented in Algorithm 3-4.

Algorithm 3-4: Sentence Similarity Measure Algorithm

```
1.  ALGORITHM( SSM )
2.  Input:
3.  s1,s2, textfile1, textfile2:String
4.  WordNet: AmhWordNet
5.  Variables:
6.  B1:Boolean=False
7.  B2:Boolean=False
8.  Wordsimscore:List
9.  SemRe:Float
10. Output:
11. Sentence similarity score
12.  BEGIN
13.        FOR EACH Word_m IN s1
14.            FOR EACH Word_n IN s2
15.            IF Word_m IN textfile1
16.               READ Best_sense1
17.               CID1 = Best_sense1
18.               B1=True
19.            END IF
20.            IF Word_n IN textfile2
21.               READ Best_sense2
22.               CID2 = Best_sense2
23.               B2=True
24.            END IF
25.            IF POS of Word_m is equal to POS of  Word_m
26.               IF Word_m is equal to Word_n
27.                  IF B1 == True OR B2 == True
28.                     IF CID1 == CID2
29.                        APPEND 1 TO  Wordsimscore
30.                     ELSE
31.                        APPEND 0 TO Wordsimscore
32.                     END IF
33.                  ELSE
34.                     APPEND 1 TO  Wordsimscore
35.               ELSE
36.                   IF B1 is True
37.                      WCID1 = CID1
38.                   ELSE
39.                       READ Word_m,ConceptId FROM AmhWordNet
40.                       WCID1 = ConceptId
41.                   END IF
42.                   IF B2 is True
43.                      WCID1 = CID1
```

```
44.                     ELSE
45.                         READ Word_n,ConceptId FROM AmhWordNet
46.                            WCID2 = ConceptId
47.                     END IF
48.                     IF WCID1 == WCID2
49.                        APPEND 1 TO  Wordsimscore
50.                     ELSE
51.                         SemSim = SemanticSimilarity(WCID1,WCID2)
    // SemSim is the output of Algorithm 3-3that calculate the Semantic
    similarity between concepts
52.                             APPEND SemSim TO Wordsimscore
53.                     END IF
54.                   END IF
55.                ELSE
56.                   APPEND 0 TO  Wordsimscore
57.                END IF
58.           END FOR
59.           APPEND Wordsimscore TO wordMatrix
60.           Wordsimscore = []
61.       END FOR
62.       FOR EACH  wordMa IN wordMatrix
63.                 Wordsim=MAXIMUM(wordMa)
64.                 APPEND   Wordsim TO Total
65.                 Sensim =2* SUM(Total)/(length(s1)+length(s2))
66.       END FOR
67. END
68. END ALGORITHM(SSM)
```

Algorithm 3-4 shows how similarity between sentences is computed. The algorithm takes two sentences and each sentence words similarity is computed. The algorithm also takes two text files which are the output of Algorithm 3-2 (WSD Algorithm). The text files have ambiguity words of the documents with their sense. During computing similarity between words the flow which is shown in Figure 3-5 is implemented which is, POS of the words are identified first and if they are different the similarity score will be 0 otherwise the first task is to identify whether the words are the same or not. If they are the same, the words are identified whether they are ambiguous or not using text file1 and text file 2. If they are ambiguous they will be available in the texts files with correct senses and if they have the same sense (their concept id is the same) the similarity score will be 1 otherwise it will be 0. If they are not ambiguous the score is 1. If the words are not the same, the next task will be finding their concept Id from AmhWordNet and if they are on the same concept they are similar and the similarity score will be 1 else finding their semantic similarity using Wu and Palmer will be done through their concept ids on line 51. Finally the similarity between sentences is computed using average matching formula on line 65. The procedure which is listed above also applied in order to get sentence similarity score.

Document Similarity Score Module

The result of sentence similarity score is used to calculate the overall document similarity. The maximum score of sentences of document 2 with sentences of document 1 is taken and using Equation 3-1 document similarity score is done.

$$Sim(D1,D2) = \frac{\sum_{si \epsilon D1, sj \epsilon D2} MaxSim(si,sj)}{|D1|} \quad \textbf{3-1}$$

Where,

D1- Document 1

D2- Document 2

S_i -Sentences of D1

S_j -Sentences of D2

|D1|- Length of D1

The algorithm for over all Document similarity measure is shown in Algorithm 3-5.

Algorithm 3-5: Document Similarity Measure Algorithm

```
1.  ALGORITHM(DSM)
2.  Input:
3.  Documents: Doc1,Doc2
4.  WordNet: AmhWordNet
5.  Variables:
6.  B1:Boolean=False
7.  B2:Boolean=False
8.  Wordsimscore:List
9.      Sen:List
10. SemRe:Float
11. Output:
12.     Document similarity score
13.   BEGIN
14. READ Doc1,Doc2
15. FOR EACH Sentence_m IN Doc1
16.       FOR EACH Sentence_n IN Doc2
17.           senscore=SSM( Sentence_m,Sentence_n  )
18.           Append senscore TO Scorelist
19.       END FOR
20.         Append MAX(Scorelist) IN Sen
21. END FOR
22. L1=Length(Doc1)
23. L2=Length(Doc2)
24. DocSimilarity=SUM(Sen)/L1
25.  END
26. END ALGORITHM(DSM)
```

Algorithm 3-5 is responsible to compute similarity between documents based on the sentence similarity which is implemented in Algorithm 3-4. First the similarity between sentences is computed then the document similarity is calculated using Equation 3-1.

CHAPTER FOUR

EXPERIMENT

4.1 Overview

In this chapter experiments conducted to evaluate the performance of the proposed approach are presented. Experimental results are done between clusters made by human and systems from 40 datasets (documents).The names of clusters are listed in Table 4-1.The clusters which are made by systems are formed from the results of similarity score among the 40 documents measured using CADS, CADSWoWSD (CADS without WSD), PMI, Jaccard and Cosine similarity measures. The reason why we select the similarity measures cosine and Jaccard from traditional and PMI from statistical approaches are; first they are simple to implement and they are the common ones and second, we want to show the performance of similarity measures of different approaches on this experiment. The CADSWoWSD also implemented to prove the influence of WSD in a similarity measure.

The environment used to conduct this experiment is Acer Laptop having 3GB RAM, duo core processor and Window7 64-bit Operating System. In next sub-sections, we will discuss the procedure followed to conduct experiments and its result.

4.2 Experimental Procedure

This section describes the preparations made to carry out experiments that test the performance of CADS.

4.2.1 Data Collection

To carry out the experiments, two types of datasets are required; dataset to build the AmhWordNet and dataset to test the performance of CADS. The first data set is collected from Amharic Dictionary [**20**]. Words with their POS and concept definition are the information used from the dataset in order to build AmhWordNet. The AmhWordNet has 500 different words, 507

concepts and 554 senses of words. The second dataset is a collection of 40 sport news collected from Walta Information Center (WIC). Sample sport news articles are attached as Appendix A.

4.2.2 Manual Document Clustering

The 40 Amharic sport news documents were manually grouped into 9 clusters. These clusters are used to measure the performance of our system. Table 4-1 shows the name of clusters and Table 4-2 shows the resulting 9 clusters.

Table 4-1 : Name of Clusters

No.	Cluster Name	Cluster Code
1	አለም አቀፍ ሩጫ (International Athletics)	C1
2	ስታዲየም ግንባታ (Stadium construction)	C2
3	አትሌት ሀይሌ ገ/ስላሴ (Athlete Haile G/Silase)	C3
4	የኢትዮጵያ ፕሪሚየርሊግ (Ethiopian Premier League)	C4
5	የኢትዮጵያ እግርኳስ ፌዴሬሽን(Ethiopian Football Federation)	C5
6	የአትሌቶች አስተያየት በእግርኳስ(Athletes Comment on Football)	C6
7	የአፍሪካ እግርኳስ(African Football)	C7
8	አለም አቀፍ እግርኳስ ክለቦች(International Football Clubs)	C8
9	የኢትዮጵያ ብሄራዊ ሊግ(Ethiopian National League)	C9

Table 4-2: Human Based clusters of Documents

No.	Cluster Code	Documents
1	C1	D1,D2,D3,D4,D15,D16,D17,D19,D20
2	C2	D30,D36
3	C3	D5,D6,D18
4	C4	D7,D8,D11,D21,D26,D31,D34,D35
5	C5	D9,D10,D12,D33,D39
6	C6	D13,D14
7	C7	D23,D25,D37,D38,D40
8	C8	D22,D24,D28
9	C9	D27,D29,D32

4.3 Evaluation

To evaluate the performance of CADS, first we have implemented CADSWoWSD, PMI, Jaccard and Cosine similarity measures. Then evaluation of these five similarity approaches is done using precision, recall and harmonic mean of recall and precision (f-measure) that compare clusters made from the result of document similarity scored by CADS, CADSWoWSD, Cosine and Jaccard over human made clusters.

Document similarity scores are the results obtained from similarity computing between 40 documents. Precision is the ratio of the number of documents clustered correctly to the total number of documents in a given cluster whereas recall is the ratio of the number of documents clustered correctly and the whole documents belong to a cluster.

$$Precision = \frac{TP}{TP + FP} \quad \text{4-1}$$

Where,

TP - the number of documents which are clustered correctly in a given cluster

FP - the number of documents which are Falsely clustered in a given cluster

$$Recall = \frac{TP}{TP + FN} \quad \text{4-2}$$

Where,

TP - the number of documents which are clustered correctly in a given cluster

FN - the number of documents which are missed by a given cluster

F-measure shows the overall performance of the systems for each cluster by combining the recall and precision values.

$$F - measure = 2 * \frac{(Recall * Percision)}{(Recall + Percision)} \quad \text{4-3}$$

The cluster made from the result of the systems (CADS, CADSWoWSD, Jaccard and Cosine) is done by adapting hierarchical clustering method.

Algorithm 4-1 is used to implement clustering. Before implementing the algorithm, the maximum score that each document have with whom is taken.

Algorithm 4-1: Clustering Algorithm

```
1.  ALGORITHM(Clustering)
2.    Input:
3.  Documents:Document
4.  maximum_similarity:float
5.    Output:
6.  Clusters of documents
7.    BEGIN:
8.      READ Document D
9.      DO
10.     CREATE Cluster C FROM EACH DOCUMENTS
11.     DO
12.         FIND Nearest Cluster Ci and Cj
        //To find the nearst cluster maximum_similarity is used
13.         MERGE  Ci and Cj
14.         C=C-1
15.     UNTIL C=9
16. END ALGORITHM(Clustering)
```

Algorithm 4-1 takes list of documents and maximum similarity score of a document it has with whom. To make clusters of documents first each document are used as clusters, then find the nearest clusters and merge them. The process of finding and merging of clusters are proceed until the number of clusters become 9 as the number of clusters made by human is 9. When finding the nearest clusters the similarity score between the documents is considered.

Table 4-3 shows clusters of document made from the results of similarity scores by CADS. Table 4-4 shows clusters of document made from the results of similarity scores by CADSWoWSD. Table 4-5 shows clusters of document made from the results of similarity scores by PMI.

Table 4-6 shows clusters of document made from the results of similarity scores by cosine. Table 4-7 shows the clusters of documents made from the results of similarity score by Jaccard.

Table 4-3: Clusters of Documents Formed from the Results of CADS

No.	Documents
1	D1,D2,D3,D4,D15,D16,D17,D19,D20
2	D28,D30,D36
3	D5,D6,D18
4	D7,D8
5	D9,D10,D12,D23,D33,D38,D40
6	D11,D26,D25,D34,D35
7	D13,D14
8	D21,D22,D24,D27,D29,D31,D32
9	D37,D39

Table 4-4: Clusters of Documents Formed from the Results of CADSWoWSD

No.	Documents
1	D1,D2,D15,D16,D17,D19,D20
2	D3,D4
3	D5,D6,D18
4	D7,D8
5	D9,D10,D12,D23,D28,D30,D33,D36,D38,D40
6	D11,D26,D25,D35
7	D13,D14
8	D21,D22,D24,D27,D29,D31,D32,D34
9	D37,D39

Table 4-5: Clusters of Documents Formed from the Results of PMI

No.	Documents
1	D1,D2,D19,D20
2	D3,D4
3	D5,D6,D15,D16,D17,D18
4	D7,D8
5	D9,D10,D12,D28,D30,D33,D36,D38,D40
6	D11,D25,D26,D35
7	D13,D14
8	D21,D22,D24,D27,D29,D31,D32
9	D23,D37,D39

Table 4-6: Clusters of Documents Formed from the Results of Cosine

No.	Documents
1	D1,D2, D19,D20
2	D3,D4
3	D5,D6,D15,D16,D17,D18
4	D7,D8,D32
5	D9,D30,D33,D36,D37
6	D10,D12,D23,D25,D28,D31
7	D11,D26,D35,D39
8	D13,D14
9	D21,D22,D24,D27,D29,D34,D38,D40

Table 4-7: Clusters of Documents Formed from the Results of Jaccard

No.	Documents
1	D1,D2, D19,D20
2	D3,D4
3	D5,D6
4	D7,D8,D22,D24,D32,D33,D34
5	D9,D10,D12,D23,D28,D30,D36,D37
6	D11,D25,D26,D35,D39
7	D13,D14
8	D15,D16,D17,D18
9	D21,D27,D29,D31,D38,D40

4.4 Result

As we have discussed above, the experiment is done between the five systems over human judgment. Precision and recall values are computed for all the clusters of five systems. The results are attached as Appendix B.

The chart presented in Figure 4-1 shows the precision values of each cluster for the five systems.

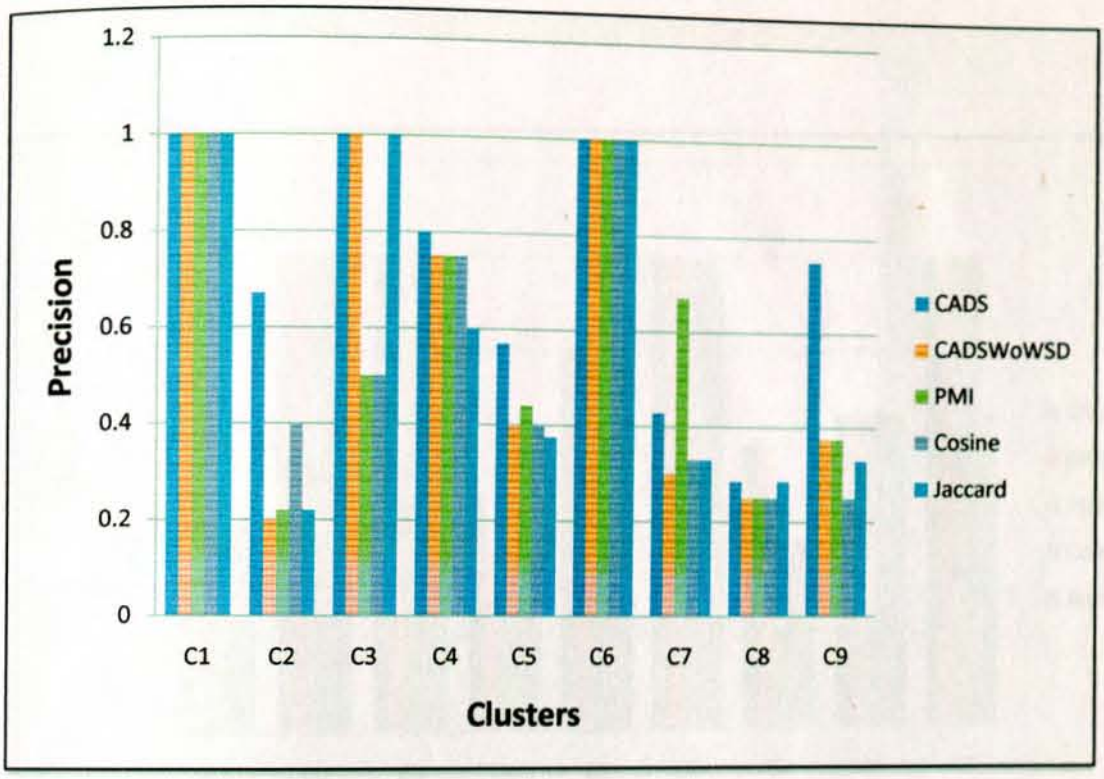


Figure 4-1:Precision Values for all the Clusters for CADS, CADSWoWSD, PMI, Jaccard and Cosine

When we compare the precision values for the CADS, as shown in Figure 4-1, the precision has increased for "C2", "C4", "C5", and "C9".For instance when we look at "C2", 2/3 of documents are grouped together in this cluster are correct for CADS whereas in the other systems from the correctly clustered documents are very few. All systems have a precision value 1 for "C1" and "C6". This indicates, all the documents intended to group in these clusters are correct and no incorrect documents are there in the clusters.

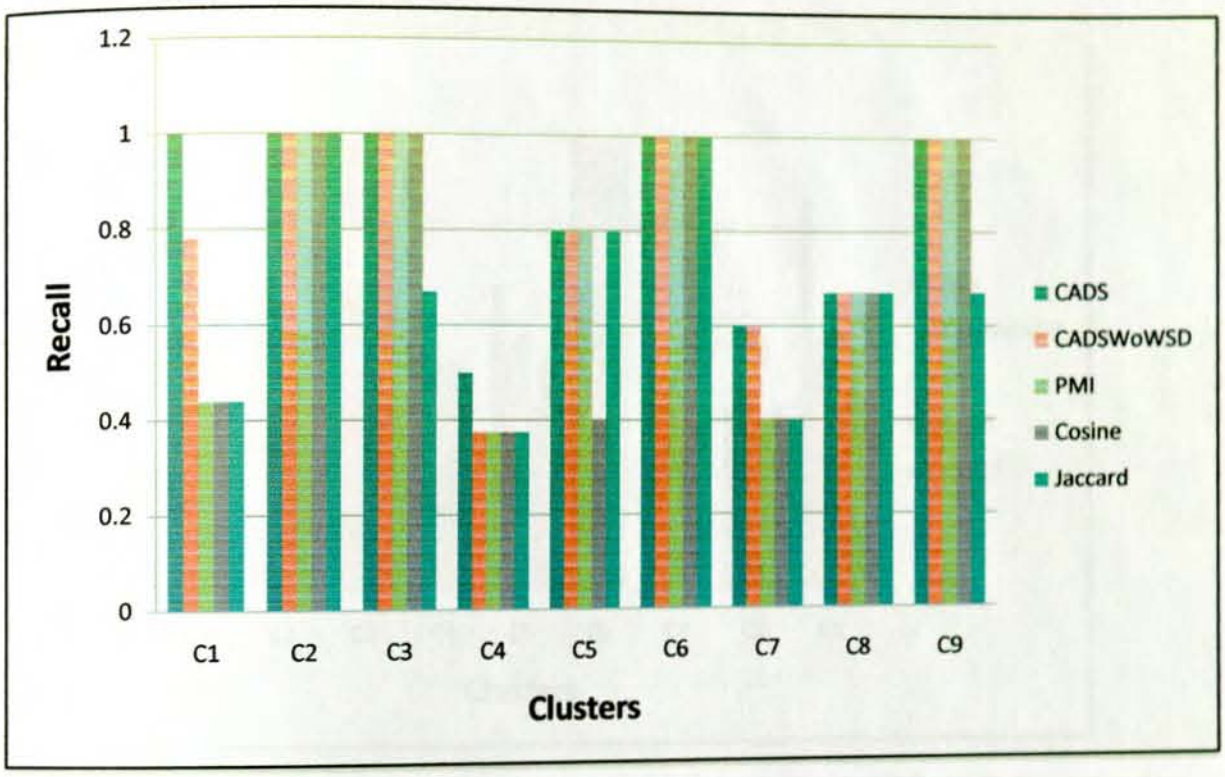


Figure 4-2: Recall Values for Clusters of CADS, CADSWoWSD, PMI, Cosine and Jaccard

Figure 4-2 shows recall values computed for each clusters for five systems. As presented in the graph, CADS has better recall values compared to the other measures. For example, when we look at "C1", all documents intended to group together in the cluster are grouped together using CADS while the other systems miss some of the documents in the cluster. The clusters which are formed by CADS have better recall values compared to the other four systems. Out of nine clusters, five of them have recall value 1.

In order to show a better view of the performance of the systems, f-measure has been computed for all clusters of each of the systems. These f-measure values are plotted on a graph in Figure 4-3.

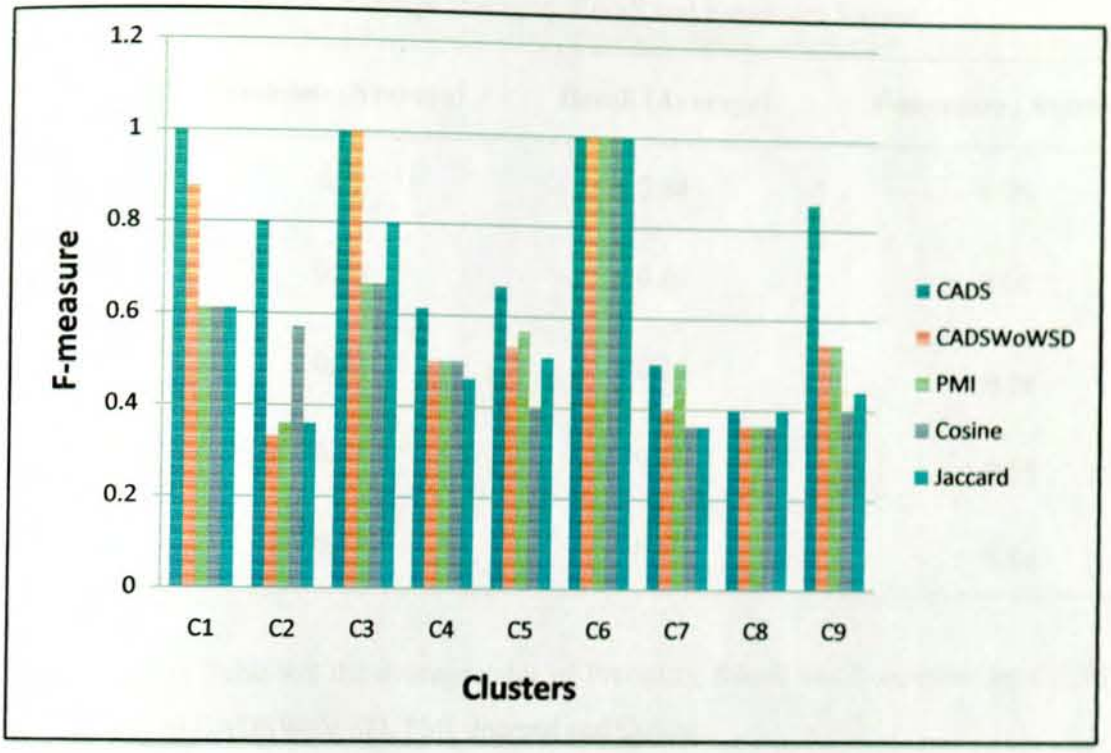


Figure 4-3: F-measure Values for All the Clusters for CADS, CADSWoWSD, PMI, Jaccard and Cosine

As shown in Figure 4-3, the f-measure values for the CADS system for "Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5" and "Cluster 9" are greater than that of the other systems. All systems have the same f-values for "Cluster 6". CADS and PMI have greater F-values for "Cluster 7" CADS and Jaccard have greater F-values for "Cluster 8". These indicate that, CADS provide accurate value compared to the other four systems.

For the 9clusters and each system, average recall, precision, and f-measure values have been calculated and it is presented in Table 4-8.

Table 4-8: Average Precision, Recall and F-measure Values

System	Precision (Average)	Recall (Average)	F-measure (Average)
CADS	0.72	0.84	0.76
CADSWoWSD	0.58	0.80	0.61
PMI	0.57	0.74	0.56
Jaccard	0.57	0.66	0.55
Cosine	0.54	0.69	0.54

As it is depicted in Table 4-8 the average value of Precision, Recall and F-measure for CADS is greater than that of CADSWoWSD, PMI, Jaccard and Cosine.

When the precision, recall and f-measure average are expressed in percentage, CADS has 72% precision, 84 % recall and 76% f-measure. CADSWoWSD has 58% precision, 80% recall and 61% f-measure. PMI has 57% precision, 74% recall and 56% f-measure. Jaccard system has 57% precision, 66% recall and 55% f-measure. Cosine system has 54% precision, 69% recall and 54% f-measure.

Figure 4-4 shows the average f-measure values of the five systems. The average f-measure indicates the overall performance of each system.

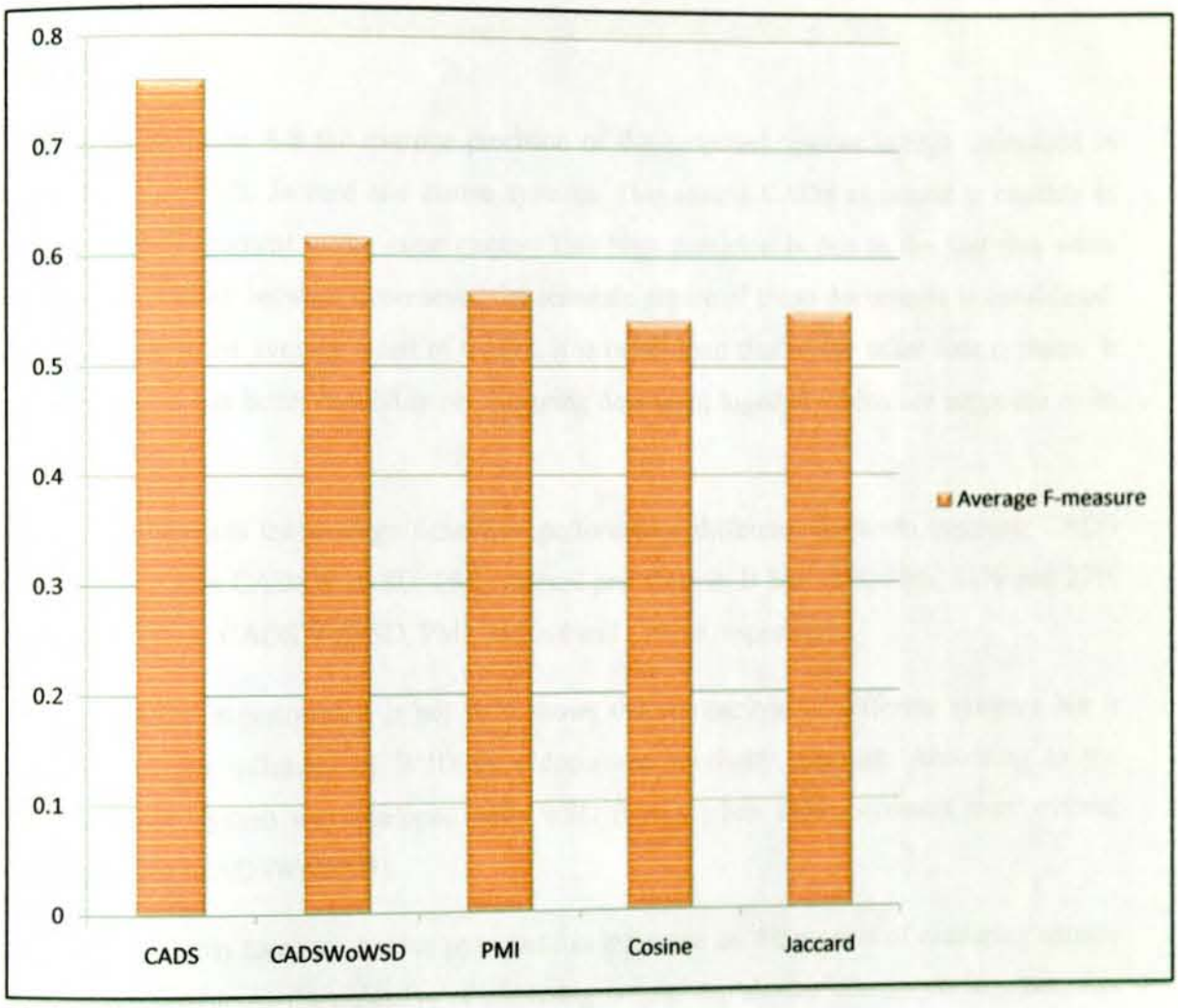


Figure 4-4: Graph of average f-measure for all Systems

As shown in Figure 4-4 documents which are clustered by CADS are closer to the human made clusters as compare to CADSWoWSD, PMI, Jaccard and Cosine.

4.5 Discussion

As depicted in Table 4-8 the average precision of the proposed system is high compared to CADSWoWSD, PMI, Jaccard and cosine systems. This means CADS clustered is capable in putting similar document in the same cluster. This high precision is due to the fact that when computing similarity between documents, the semantic nature of those documents is considered. When we look at the average recall of CADS, it is better than that of the other four systems. It point out CADS has better capability on clustering document together which are supposed to be in same cluster.

Figure 4-4 illustrates the average f-measure performance difference between systems. CADS perform better than CADSWoWSD, PMI, Jaccard and Cosine. It has 15%, 20%, 21% and 22% improvement from CADSWoWSD, PMI, Jaccard and Cosine respectively.

In the conducted experiment it is not only shows the comparison of different systems but it clearly shows the influence of WSD in a document similarity measure. According to the experiment, the system we developed with WSD (CADS) has 15% increment than without including WSD (CADSWoWSD).

Moreover, similarity measure that we proposed has influence on the process of clustering similar documents together. As the objective of clustering is to group similar documents together, this system consider not only document that are exactly the same or have the same words but it consider documents that are semantically similar (documents that are expressed differently but talk about the same things).

This research has a limitation in identifying meaning of words of documents which have word stress and the word whose meaning is known by pragmatic knowledge. If word stress and pragmatic knowledge of words are considered, our system may have better performance than this one.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis, we proposed an approach to measure document similarity considering semantic information. Different techniques and NLPs are combined in order to implement the proposed system.

The System (CADS) is composed of Pre-processing, concept based similarity measure and AmhWordNet modules. The concept based similarity module is the main part of the system where similarity between documents is measured. We implement Amharic WordNet and use it as input for CSM module. This module is a composition of sub modules such as WSD and Concept Tree Extraction. Polysemous words are disambiguated using WSD module before using them on similarity measure. Concept trees which are generated by Concept Tree Extraction are used during measuring semantic similarity between words

The proposed system was tested based on the clusters of documents made by human. Besides, the CADS is compared with the CADSWoWSD, PMI, Jaccard and Cosine similarity measures. The relevance of the generated clusters are evaluated using precision, recall and f-measure. The proposed semantic has better average recall, precision, and f-measure values compared to the systems CADSWoWSD, PMI, Jaccard and Cosine.

5.2 Contribution

The contributions of this thesis work are summarized as follows:

- A model is proposed for concept-base Amharic document similarity that takes advantage of existing semantic tool – WordNet.
- Amharic WordNet (AmhWordNet) that is domain specific to sport is implemented in order to use it for identifying words' synonym and polysemy; those are important for semantic document similarity measuring.

- Implementing WSD algorithm using AmhWordNet in order to find sense of polysemous words from documents.
- Propose an algorithm to generating Concept tree from AmhWordNet to use it for finding semantic similarity between words

5.3 Future Work

The result found in this research showed that semantic similarity between Amharic documents can be done. In the future there is a need to conduct further research on the following aspects

- This study only represents concepts in the AmhWordNet which has sport domain. All possible concepts should be represented in the WordNet in order to handle domains other than sport.
- Pragmatic knowledge is not considered in this research. The performance of the system will be better than this as knowing pragmatic knowledge has a great impact in identifying meaning of document.
- If morphological analyzer is used to eliminate prefix and suffix instead of stemmer, the performance of CADS will be better as the output of morphological analyzer is more similar with the words of AmhWordNet than the output of stemmer.
- In this research we have used sentence-based similarity measure to compute the similarity between documents. Since, words which make up the meaning of a document might sparse in a paragraph rather than being in a given sentence, we recommend exploring the advantage of paragraph-based similarity measure in order to get a better performance.

Finally, we believe that the result of this research can be modified to get better results.

Reference

[1] Acharya, S., and S. Parija. "The Process of Information Extraction through Natural Language Processing." *International Journal of Logic and Computation (IJLP)*: 40.

[2] G. *Salton* and *M. E. Lesk*. "*Computer evaluation of indexing and text processing.*" The SMART Retrieval System: Experiments in Automatic Document Processing: *143-180*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, *1971*.

[3] Dillon, Martin, and Ann S. Gray. "FASIT: A fully automatic syntactically based indexing system." *Journal of the American Society for Information Science* 34, no. 2 (1983): 99-108.

[4] JL, Fagan. "The effectiveness of a non-syntactic approach to automatic phrase indexing for document retrieval." J. Amer. Soc. Inform. Sci. 40 (1989): 115–132.

[5] Fuhr, Norbert. "Models for retrieval with probabilistic indexing." *Information Processing & Management* 25, no. 1 (1989): 55-72

[6] Griffiths, Alan, Lesley A. Robinson, and Peter Willett. "Hierarchic agglomerative clustering methods for automatic document classification." *Journal of Documentation* 40, no. 3 (1984): 175-205.

[7] Sparck Jones, Karen.*"Automatic keyword classification for information retrieval.*". London: Butterworths, 1971.

[8] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18, no. 11 (1975): 613-620.

[9] Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis." *Discourse processes* 25, no. 2-3 (1998): 259-284.

[10] Corley, Courtney, and Rada Mihalcea. "Measuring the semantic similarity of texts." In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Association for Computational Linguistics, 2005. 13-18.

[11] Chim, Hung, and Xiaotie Deng. "Efficient phrase-based document similarity for clustering." *Knowledge and Data Engineering, IEEE Transactions on* 20, no. 9 (2008): 1217-1229.

[12] Steinberger, Ralf, Bruno Pouliquen, and Johan Hagman. "Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc." In *Computational Linguistics and Intelligent Text Processing*, Springer Berlin Heidelberg, 2002. 415-424.

[13] Mulugeta Bayeh, "Text retrieval using self-organized document map: The case of ILRI digital library," Master's Thesis, Addis Ababa University, June, 2002.

[14] Tewodros Hailemeskel, "Amharic Text Retrieval: An Exprement using Latent Semantic Indexing with Singular Value Decomposition," Master's Thesis, Addis Ababa University, 2003.

[15] Argaw, Atelach Alemu, Lars Asker, Rickard Cöster, and Jussi Karlgren, "Dictionary-based Amharic - English Information Retrieval," In Proc. Cross Language Evaluation Forum (CLEF 2004).

[16] Argaw, Atelach Alemu, Lars Asker, Rickard Cöster, and Jussi Karlgren. "Dictionary-based Amharic–English information retrieval." In *Multilingual Information Access for Text, Speech and Images*, Springer Berlin Heidelberg, 2005. 143-149

[17] Surafel Teklu, "Automatic Categorization of News Text: A Machine Learning Approach," Master's Thesis, Addis Ababa University, July, 2003.

[18] Eyassu, Samuel, and Björn Gambäck. "Classifying Amharic news text using self-organizing maps." In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Association for Computational Linguistics, 2005.71-78.

[19] Meron Sahlemariam, Mulugeta Libsie, Daniel Yacob."Concept-Based Automatic Amharic Document Categorization." AMCIS *Proceeding* 2009 : 116

[20] የኢትዮጵያ ቋንቋዎች ጥናትና ምርምር ማእከል, *አማርኛ መዝገበ ቃላት*. አዲስአበባ: አርቲስቲክ ማተሚያ ቤት, 1993.

[21] Lin, Chin-Yew, and Eduard Hovy. "Automatic evaluation of summaries using n-gram co-occurrence statistics." In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*- Association for Computational Linguistics, 1(2003). 71-78.

[22] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation." In *Proceedings of the 40th annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2002. 311-318.

[23] Islam, Aminul, and Diana Inkpen. "Semantic text similarity using corpus-based word similarity and string similarity." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2, no. 2 (2008): 10.

[24] Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. "WordNet:: Similarity: measuring the relatedness of concepts." In *Demonstration Papers at HLT-NAACL*. Association for Computational Linguistics, 2004. 38-41.

[25] Leacock, Claudia, and Martin Chodorow. "Combining local context and WordNet similarity for word sense identification." *WordNet: An electronic lexical database* 49, no. 2 (1998): 265-283.

[26] Wu, Zhibiao, and Martha Palmer. "Verbs semantics and lexical selection." In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1994. 133-138.

[27] Resnik, Philip. "Using information content to evaluate semantic similarity in a taxonomy." *arXiv preprint cmp-lg/9511007* (1995).

[28] Hirakawa, Hideki, Zhonghui Xu, and Kenneth Haase. "Inherited Feature-based Similarity Measure based on large semantic hierarchy and large text corpus." In *Proceedings of the 16th conference on Computational linguistics*. Association for Computational Linguistics, *1*(1996), 508-513.

[29] P. Turney, "Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of the Twelfth European Conference on Machine Learning.," ECML-2001.

[30] Church, Kenneth Ward, and Patrick Hanks. "Word association norms, mutual information, and lexicography." *Computational linguistics* 16, no. 1 (1990): 22-29.

[31] Castillo, Julio J., and Marina E. Cardenas. "Using sentence semantic similarity based on WordNet in recognizing textual entailment." In *Advances in Artificial Intelligence–IBERAMIA*. Springer Berlin Heidelberg, 2010. 366-375.

[32] Lee, Lillian. "Measures of distributional similarity." In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 1999. 25-32

[33] Zhang, Junsheng, Yunchuan Sun, Huilin Wang, and Yanqing He. "Calculating statistical similarity between sentences." *Journal of Convergence Information Technology* 6, no. 2 (2011).

[34] Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. "Indexing by latent semantic analysis." *JASIS* 41, no. 6 (1990): 391-407.

[35] Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38, no. 11 (1995): 39-41.

[36] Gabrilovich, Evgeniy, and Shaul Markovitch. "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis." In *IJCAI*, 7(2007).1606-1611.

[37] Zequan LIU , Qian DONG Dingjia LIU, "A Dependency Grammar and WordNet Based Sentence," *Journal of Computational Information Systems* 8, no. 3 (2012): 1027-1035.

[38] Strube, Michael, and Simone Paolo Ponzetto. "WikiRelate! Computing semantic relatedness using Wikipedia." *In Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 2,Menlo Park, CA; Cambridge, MA; London; Press, AAAI; MIT, 21 , no.2(2006):1419-1424.

[39] Witten, I., and David Milne. "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links." In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, 2008. 25-30.

[40] Li, Yuhua, David McLean, Zuhair A. Bandar, James D. O'shea, and Keeley Crockett. "Sentence similarity based on semantic nets and corpus statistics." *Knowledge and Data Engineering, IEEE Transactions on* 18, no. 8 (2006): 1138-1150.

[41] Tversky, Amos. "Features of similarity." *Psychological review* 84, no. 4 (1977): 327.

[42] Accredited Language Services Blog. [Online]. http://www.alsintl.com/resources/languages/Amharic/, access on 11/3/ 2013, 10:00 am.

[43] Baye Yimam, "*አጭርና ቀላል የአማርኛ ሰዋስው*." Addis Ababa: Alpha Printers, 2002.

[44] Natural Language Processing. Articles on Natural Language Processing. [Online]. http://language.worldofcomputing.net/pos-tagging/parts-of-speech-tagging.html, access on 15/3/ 2013, 3:17 pm.

[45] Carpuat, Marine, and Dekai Wu. "Word sense disambiguation vs. statistical machine translation." In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005. 387-394.

[46] Lesk, Michael. "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone." In *Proceedings of the 5th annual international conference on Systems documentation*. ACM, 1986. 24-26.

[47] Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. "Corpus-based and knowledge-based measures of text semantic similarity." In *AAAI* 6, no.1(2006): 775-780.

[48] Ho, Chukfong, Masrah Azrifah Azmi Murad, Rabiah Abdul Kadir, and Shyamala C. Doraisamy. "Word sense disambiguation-based sentence similarity." In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010. 418-426.

[49] Xian-yi, Cheng, Sun Ping, Zhu Qian, and Cai Yue-hong. "The Research of Chinese Semantic Similarity Calculation Introduced Punctuations." *Journal of Convergence Information Technology* 5, no. 7 (2010).

[50] Ma Junhong, "A method of Phrased Integrated Semantic Similarity Computation," *Journal of Theoretical and Applied Information Technology* 49 , no. 3(2013): 825-831.

[51] Gouws, Stephan. "Evaluation and development of conceptual document similarity metrics with content-based recommender applications." PhD diss., Stellenbosch: University of Stellenbosch, 2010.

[52] Meron Sahlemariam,Teshome Kassie , Tessema Mindaye, "The Need for Amharic WordNet", The 5th International Conference of the Global WordNet Association, Mumbai, India, Febrary, 2010.

Appendix A: Sample News Articles

Document 1:

ዘንድሮ<NP>በተካሄደው<VP>የአለም<NP>ቤት<N>ውስጥ<PREP>ሻምፒዮና<N>ኢትዮጵያ<N>አመርቂ<ADJ>
ውጤት<N>አስመዝግባለች<V>።በሻምፒዮናው<NP> 172
<NUMCR>አገሮች<N>የተሳተፉ<VREL>ሲሆን<VP>
26ቱ<NUMP>ብቻ<ADV>ሜዳሊያ<N>ሠንጠረዥ<N>ውስጥ<PREP>ገብተዋል<V>ኢትዮጵያም<NP>አምስት
<NUMCR>ሜዳሊያዎችን<NP>በማግኘት<VP>በሶስተኝነት<NUMP>ውድድሩን<N>አጠናቃለች<V>።በውድድ
ሩ<NP>ወጣቱ<ADJ>መሐመድ<N>አማን<N>በአንድ<NUMP>ደቂቃ<N> 48.36
<NUMCR>ሰከንድ<N>አሸናፊ<N>በመሆ ኑ<VP>ኢትዮጵያ<N>ጎልታና<ADJC>ደምቃ<ADJ>ከነበረው<VP
>ከረዥም<ADJP>ርቀት<N>ባሻገር<PREP>በመካከለኛ<ADJP>ርቀት<N>ባለተስፋ<NP>መሆኗን<VN>አመላክ
ቷል<V>።መሐመድ<N>አምና<ADV>በዓለም<NP>ወጣቶች<N>ሻምፒዮና<N>የብር<ADJP>ሜዳሊያ<N>በማግ
ኘትም<NPC>ለአገሪቱ<NP>የመጀመሪያዋ<NP>የ800
<NUMP>ሜትር<N>ሜዳሊስት<N>ለመሆን<VP>የቻለ<VP>ነው<V>።የ3,000
<NUMP>ሜትር<N>ሴቶች<N>ሩጫ<N>በታሪክ<NP>በተከታታይ<ADV>ለአምስተኛ<NUMP>ጊዜ<N>ያሸነፉ.
ች<VREL>የመጀመርያዋ<NP>ሴት<N>ለመባል<NP>ቆርጣ<V>የነበረችው<VREL>ኮከቧ<ADJ>መሠረት<N>
ደፋር<N>ባልተጠበቀ<VP>ሁኔታ<N>በኬንያዊቷ<NP>ሄለን<N>ኦሳንዶ<N>በአንድ<NUMP>ሰከንድ<N>ተቀድ
ማ<VP>ሁለተኛ<NUMOR>ደረጃን<N>አግኝታለች<V>።በዚሁ<PRONP>ርቀት<N>ገለቴ<N>ቡርቃ<N>በሦስ
ተኛነት<NUMOR>አጠናቃለች<V>።የሴቶችን<NP> 1,500
<NUMCR>ሜትር<N>የኦሊምፒክ<NP>ድርብ<N>ወርቅ<N>ባለድሏ<NP>ጥሩነሽ<N>ዲባባ<N>እህት<N>ገንዘ
ቤ<N>ዲባባ<N>አሸንፌ<N>ስትሆን<VP>በወንዶች<NP> 1,500
<NUMCR>ሜትር<N>መኮንን<N>ገብረመድኅን<N>በሦስተኛነት<NUMOR>ነሐስ<N>ሜዳሊያ<N>አግኝቷል<
V>።

Document 2:

ኢትዮጵያ<N>በቱርክ<NP>ኢስታንቡል<N>በተካሄደው<VP>
14ኛው<NUMP>የዓለም<NP>የቤት<NP>ውስጥ<PREP>አትሌቲክስ<N>ሻምፒዮና<N>በተለያየ<ADJP>ርቀት
<N>ተሳትፌ<VREL>ሁና<VP>አስደሳች<VP>የሆነ<VP>ውጤት<N>በማግኘት<VN>ተመልሳለች<V>።ገንዘቤ<
N>ዲባባ<N>በአንድ<NUMP>ሺህ<N> 500
<NUMCR>ሜትር<N>፣<PUNC>መሐመድ<N>አማን<N>ደግሞ<CONJ>በ800
<NUMP>ሜትር<N>ያስገኟቸው<VP>የወርቅ<ADJP>ሜዳሊያዎች<N>የአገሪቱን<NP>ደረጃ<N>ከፍ<PREP>
ለማድረግ<NP>አስችለዋል<V>።መሠረት<N>ደፋር<N>በሦስት<NUMP>ሺህ<N>ሜትር<N>ሻምፒዮና<N>የብር
<NP>ሜዳሊያ<N>ስታጠልቅ<VP>መኮንን<N>ገብረመድኅን<N>በአንድ<NP>ሺህ<N>አምስት<NUMCR>መቶ
<N>ሜትር<N>ሶስተኛ<NUMCR>ደረጃን<NP>አስመዝግበዋል<V>።እንዲሁም<PRONP>ገለቴ<N>ቡርቃ<N>
<N>ሜትር<N>ሶስተኛ<NUMCR>ደረጃን<NP>አስመዝግበዋል<V>።እንዲሁም<PRONP>ገለቴ<N>ቡርቃ<N>
በሦስት<NUMP>ሺህ<N>ሜትር<N>ያመጣችው<V>የነሐስ<N>ሜዳሊያ<N>አገሪቱ<NP>በሻምፒዮናው<NP>ላገ
ኘችው<VP>ደረጃ<N>አስተዋጽኦ<N>አድርጓል<V>።በአጠቃላይ<NP>የሜዳሊያ<NP>ድምር<N> 2
<NUMCR>ወርቅ<N>፣<PUNC> 1 <NUMCR>ብርና<NP> 2
<NUMCR>ነሐስ<N>በማግኘት<NP>ኢትዮጵያ<N>ከአፍሪካ<NP>ቀዳሚውን<NP>ስፍራ<N>ስትይዝ<VP>በዓ
<NUMCR>ለም<NP>አቀፍ<N>ደረጃ<N>ደግሞ<CONJ>ከአሜሪካና<NP>ከእንግሊዝ<NP>ቀጥላ<N>የሦስተኛ<NUMP>ደ
ረጃ<N>የሚያስገኙላትን<VREL>ውጤት<N>አስመዝግባለች<V>።ጎረቤት<N>ሀገር<N>ኬንያ<N>ሁለት<NUMC
R>የወርቅ<NP>፣<PUNC>አንድ<NUMCR>የብርና<NP>አንድ<NUMCR>የነሐስ<NP>ጨምሮ<V>አራት<N
UMCR>ሜዳሊያዎችን<NP>በማግኘት<VP>የአራተኛ<NUMP>ደረጃ<N>አግኝታለች<V>።

Document 3:

ባለፍው<ADV>ወር<N>በተካሄደው<NP>የአፍሪካ<NP>እግርኳስ<N>የማጣሪያ<NP>ጨዋታ<N>በኢትዮጵያው<NP>ተጋጣሚው<N>ላይ<PREP>ከስነምግባር<NP>ውጭ<N>የተተናኮሰውን<VP>የናሚቢያው<NP>አምበል<N>ከስድስት<NP>ጨዋታዎች<N>እንዲታገድ<VP>መወሰኑን<VP>የአፍሪካ<NP>እግርኳስ<N>ፌዴሬሽን<N>አስታወቀ<V>::የፊረንሳይ<NP>ዜና<NP>አገልግሎት<N>ከዊንድሆክ<NP>እንደዘገበው<VP>ውሳኔው<VP>የተላለፈው<VP>ሄንሪክ<N>ቦትስ<N>የተባለው<VP>የናሚቢያ<NP>አምበል<N>በተጋጠሚው<NP>የኢትዮጵያዊው<NP>ቡድን<N>ላይ<PREP>በመትፋቱ<VP>ነው<V>::

Document 4:

በቤንችማጂ<N>ዞን<N>የሚዛን<NP>ከተማ<N>አስተዳደር<N>ከ1 <N>ነጥብ<N> 4 <N>ሚሊዮን<N>ብር<N>በላይ<PREP>በሆነ<AUX>ወጪ<N>የስፖርት<NP>ማዘውተሪያና<NP>የወጣቶች<NP>ማዕከል<N>ለማሰራት<NP>ዝግጅት<N>ማጠናቀቁን<VP>ገለጸ<V>::

በአስተዳደሩ<NP>የወጣቶችና<NP>ስፖርት<N>ጽህፈት<N>ቤት<N>የስፖርት<NP>ቡድን<N>መሪ<N>አቶ<N>ዳዊት<N>ገመ<N>ለኢትዮጵያ<NP>ዜና<N>አገልግሎት<N>እንደገለጹት<VP>በሚዛንና<NP>በአማን<NP>ከተሞች<NP>ለሚገኙ<VP>ወጣቶች<N>የስፖርት<NP>ማዘውተሪያ<N>የወጣቶች<N>መዝናኛ<N>ማዕከልና<NP>ቤተመጻህፍት<N>ለመስራት<NP>ዝግጅት<N>ተጠናቋል::

በአማን<NP>ከተማ<N>ለሚሰራው<VP>የስፖርት<NP>ማዘውተሪያ<N>ስፍራ<N>የቦታ<NP>ርክክብ<N>መደረጉንና<NC>ሀብረተሰቡም<N>የጉልበት<N>ድጋፍ<N>ለማድረግ<VP>መዘጋጀቱን<NP>የቡድን<NP>መሪው<N>ተናግረዋል<V>::በሚዛን<NP>አጠቃላይ<N>ሁለተኛ<ADJ>ደረጃ<N>ትምህርት<N>ቤት<N>ውስጥም<PREP>የኳስ<N>መጫወቻ<N>ሜዳ<N>እንደሚሰራ<VP>ገልጸዋል<V>::እንዲሁም<PRONP>ወጣቶች<N>የአረፍት<N>ጊዜያቸውን<NP>የሚያሳልፉበት<VP>የመዝናኛ<NP>ማዕከል<N>እንደሚገነባ<VP>ተናግረዋል<V>::

Document 5:

በአማራ<NP>ክልል<N>አሮሚያ<N>ዞን<N>የከሚሴ<NP>ከተማ<N>አስተዳደር<N>የእግርኳስ<NP>ሜዳ<N>ለማሰራት<VP>የሚያስችል<NP> 200 <N>ሺህ<NP>ብር<N>መመደቡን<VREL>የዞኑ<NP>ወጣቶችና<N>ስፖርት<N>መምሪያ<N>አስታወቀ<V>::

የመምሪያው<N>ኃላፊ<N>አቶ<N>ቦጋለ<N>ዓለሙ<N>የማስ<N>ስፖርት<N>አሰልጣኞች<N>የተሳተፉበት<NP>ስልጠና<N>ትናንት<ADV>በከሚሴ<NP>ከተማ<N>በተጀመረበት<VP>ወቅት<N>እንደተናገሩት<VP>ወጣቶችን<N>በስፖርት<NP>በማጎልበት<VP>በልማት<NP>ስራዎች<N>ንቁ<N>ተሳታፊ<N>እንዲሆኑ<VP>ለማስቻል<NP>የስፖርት<NP>ሜዳው<N>መሰራት<VN>ከፍተኛ<N>አስተዋጽኦ<N>ይኖረዋል<V>::

የስፖርት<NP>ሜዳው<N>ለወደፊቱ<NP>ሁለገብ<N>ስታዲየም<N>ለመገንባት<VP>በሚያስችል<VP>መልኩ<N>የሚሰራ<VP>መሆኑን<VN>የተናገሩት<VP>ኃላፊው<N>በአሁኑ<NP>ወቅት<N>የጥናትና<NP>ዲዛይን<N>ስራ<N>ተጠናቆ<V>የሜዳው<NP>ጠረጋ<V>ስራና<NP>የከብር<NP>ትሪቡኑ<NP>ግንባታ<V>ለመጀመር<NP>ዝግጅት<N>እየተደረገ<VP>መሆኑን<VN>አስታውቀዋል<V>::

Appendix B: The precision, recall, and f-measure values for each of the clusters for five systems

Clusters	CADS					
	TP	FP	FN	P	R	F-M
Cluster1	9	0	0	1	1	1
Cluster2	2	1	0	0.67	1	0.80
Cluster3	3	0	0	1	1	1
Cluster4	4	1	4	0.8	0.5	0.615
Cluster5	4	3	1	0.57	0.8	0.665
Cluster6	2	0	0	1	1	1
Cluster7	3	4	2	0.428	0.6	0.499
Cluster8	2	5	1	0.285	0.67	0.399
Cluster9	3	4	0	0.75	1	0.857
Clusters	**CADSWoWSD**					
	TP	FP	FN	P	R	F-M
Cluster1	7	0	2	1	0.78	0.876
Cluster2	2	8	0	0.2	1	0.33
Cluster3	3	0	0	1	1	1
Cluster4	3	1	5	0.75	0.375	0.5
Cluster5	4	6	1	0.4	0.8	0.53
Cluster6	2	0	0	1	1	1
Cluster7	3	7	2	0.3	0.6	0.4
Cluster8	2	6	1	0.25	0.67	0.36
Cluster9	3	5	0	0.375	1	0.54
Clusters	**PMI**					
	TP	FP	FN	P	R	F-M
Cluster1	4	0	5	1	0.44	0.61
Cluster2	2	7	0	0.22	1	0.36
Cluster3	3	3	0	0.5	1	0.66

Cluster4	3	1	5	0.75	0.375	0.5
Cluster5	4	5	1	0.44	0.8	0.567
Cluster6	2	0	0	1	1	1
Cluster7	2	1	3	0.67	0.4	0.5
Cluster8	2	6	1	0.25	0.67	0.36
Cluster9	3	5	0	0.375	1	0.54
	Jaccard					
Clusters	TP	FP	FN	P	R	F-M
Cluster1	4	0	5	1	0.44	0.61
Cluster2	2	7	0	0.22	1	0.36
Cluster3	2	0	1	1	0.67	0.802
Cluster4	3	2	5	0.6	0.375	0.46
Cluster5	3	5	2	0.375	0.8	0.51
Cluster6	2	0	0	1	1	1
Cluster7	2	4	3	0.33	0.4	0.36
Cluster8	2	5	1	0.285	0.67	0.399
Cluster9	2	4	1	0.33	0.67	0.44
	Cosine					
Clusters	TP	FP	FN	P	R	F-M
Cluster1	4	0	5	1	0.44	0.611
Cluster2	2	3	0	0.4	1	0.57
Cluster3	3	3	0	0.5	1	0.66
Cluster4	3	1	5	0.75	0.375	0.5
Cluster5	2	3	3	0.4	0.4	0.4
Cluster6	2	0	0	1	1	1
Cluster7	2	4	3	0.33	0.4	0.36
Cluster8	2	6	1	0.25	0.67	0.36
Cluster9	2	6	1	0.25	1	0.4

Appendix C: Amharic Stop Words List

እዚህ	ሌላ	እባክሽ
እዚያ	ሌሎች	እባክህ
ከ	ሁሉ	ተጨማሪ
ናቸው	እያንዳንዱ	ውጪ
ትናንት	እያንዳንዳቸው	ናት
ጥቂት	እንዲሁም	ነበሩ
በርካታ	እንደገና	ነበረች
ብቻ	ማንም	ያ
ሁሉም	እባክዎ	ነገሮች
ከፊት	ከላይ	ታች
ከታች	በታች	የታች
ከውስጥ	በውስጥ	የውስጥ
ኋላ	ከኋላ	መካከል
ከመካከል	ሰሞኑን	ከሰሞኑ
በሰሞኑ	የሰሞኑ	ጋራ
የጋራ	ከጋራ	ተለያዩ
ተለያየ	ድረስ	እስከ
በጣም	ግን	ሲሆን
ሲል	ውስጥ	ላይ
ነይ	ነው	ጋር
ናቸው	ይህ	ወደ
ወዘተ	እና	ወይም
እንደ	ፊት	ወደፊት
ነገር	በብላ	በኩል
ስለ	ደግሞ	እንጂ

Declaration

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

Declared by:

Name: **Addisalem Abera Wordofa**

Signature: _____

Date: 30/12/13

Confirmed by advisor:

Name: **Fekade Getahun(Ph.D)**

Signature: _____

Date: 30/12/13

Place and date of submission: Addis Ababa University, December, 2013.