



Jimma University
Jimma Institute of Technology
School of Graduate Studies
Department of Information Technology

**Classification of Multilingual Under-resourced
Language Documents using English Ontology**

Tsegay Mulu Kassa

A Thesis submitted to the School of Graduate Studies of Jimma University in
partial fulfillment of the requirements for the degree of Master of Science in
Information Technology

Jimma, Ethiopia
November 2017


Jimma University
Jimma Institute of Technology
School of Graduate Studies
Classification of Multilingual Under-resourced Language Documents using
English Ontology

Tsegay Mullu Kassa

Advisor: Dr. Yaregal Assabie

This is to certify that the thesis prepared by *Tsegay Mullu Kassa*, titled: *Classification of Multilingual Under-resourced Language Documents using English Ontology* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Technology complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: <u>Yaregal Assabie</u>		<u>11/13/2017</u>
Co- advisor: _____		
Examiner: _____		
Examiner: _____		

Abstract

Automatic documents classification is an important task due to the rapid growth of the number of electronic documents. Classification aims to assign the document to a predefined category automatically based on its contents. In general, text classification plays an important role in information extraction and summarization, text retrieval, question answering, e-mail spam detection, web page content filtering, and automatic message routing. Most existing methods and techniques in the field of document classification are keyword based without many features. Due to lack of semantic consideration of this technique it is outperformed by ontology based text categorization approach. However, it is very challenging of building ontology with under-resourced language, ontology-based classification is limited to English language support. Hence, under-resourced written documents are not benefited from ontology based text classification.

In this research, we propose an approach that can classify under-resourced language written documents on top of resourced language ontology. Beside this, the proposed approach also is capable of classifying multilingual documents (i.e. Amharic, Afaan Oromo and Tigrinya textual documents) on top of English ontology. Furthermore, in order to show the practicality of the proposed approach a prototype is developed using a java framework. To evaluate the performance of the proposed approach 20 test documents for Amharic and Tigrinya and 15 test document for Afaan Oromo in each news category is used. In order to observe the effect of incorporated features (i.e. lemma based index term selection, pre-processing strategies (i.e. stopword removal and stemming) during concept mapping and semantical based concept mapping) in the proposed document classifier four experimental techniques were conducted. The experiments were evaluated using Recall, Precision and F-measure in order to observe the impact of the proposed approach in the improvement of document classification process. The experimental results show that the proposed document classifier with incorporation of all features and components achieved the average F-measure of 92.37%, 86.07% and 88.12% for Amharic, Afaan Oromo and Tigrinya documents respectively. These results proved that the proposed approach contributes effectively in the process of classifying under-resourced written documents (i.e. Amharic, Afaan Oromo and Tigrinya documents) on top of resourced language ontology (i.e. English ontology). To enhance the effectiveness of the proposed approach the researcher recommends enhancing the size and quality of bilingual dictionary, and enhancing the performance of part of speech tagging and morphological analyzer.

Keywords: Multilingual, Text Mining, Documents or text Classification, News Ontology, knowledge base, Ontology based text categorization, keyword based, Multilingual text classification, Ontology.

Dedicated to
My mom, Etenesh Brhanu Beyene
&
My lovely, kido

Acknowledgment

First of all, I thank God for giving me the strength and courage to persevere throughout the duration of this thesis and made all of this and everything else possible.

I am deeply grateful to my advisor Dr. Yaregal Assabie and my co-advisor Mr. Teferi Kebebew for their continued encouragement, unlimited efforts, persistent motivation, support, and great knowledge throughout my thesis, without their help, guidance, and follow-up, this thesis would never have been possible.

I am greatly indebted to my family: my dear father, my dear mother, my wife, and my brothers and sister for their encouragement and support during my studies and during my thesis work.

Last but not least, I extend my thanks to all my friends.

Table of Contents

List of Figures	vii
List of Tables	viii
List of Algorithms	ix
List of Acronyms	x
Introduction.....	1
1.1 Overview	1
1.2 Statement of the Problem	4
1.3 Objectives.....	5
1.3.1 General Objective	5
1.3.2 Specific Objectives	5
1.4 Methodology	6
1.4.1 Literature Review.....	6
1.4.2 Data Collection	6
1.4.3 Design the Framework.....	6
1.4.4 Prototype Development	6
1.4.5 Evaluation	7
1.5 Scope and Limitation	7
1.6 Application of Results.....	7
1.7 Thesis Organization.....	8
Literature Review.....	9
2.1 Introduction	9
2.2 Text Categorization	9
2.2.1 Text Categorization based on Application Area.....	11
2.2.2 Text Categorization based on Usage.....	12
2.2.3 Text Categorization Automation.....	12
2.3 Steps in Automatic Text Classification.....	13
2.3.1 Pre-processing.....	13
2.3.2 Classification.....	16

2.3.3	Performance Evaluations	20
2.4	Ontology.....	21
2.4.1	Ontology Building	23
2.4.2	Ontology Building Tools	24
2.4.3	Ontology Building Languages	25
2.4.4	Ontology Evaluation	25
2.5	Text Classification and Multilinguality	26
2.6	Under-resourced languages	28
2.7	Summary	29
Related Work	30
3.1	Introduction.....	30
3.2	Machine Learning Approach to Text Classification	30
3.3	Knowledge Based Approach to Text Classification	31
3.4	Text Classification and Multilinguality	33
3.5	Summary	35
Design of Multilingual Classifier for Under-resourced Language Documents	37
4.1	Introduction	37
4.2	Tools used	37
4.2.1	TreeTagger	37
4.2.2	HornMorpho	38
4.2.3	Jena	40
4.2.4	Snowball Porter Stemmer	41
4.2.5	Wordnet.....	42
4.3	Architecture of Multilingual Documents Classifier	45
4.4	Pre-processing	46
4.4.1	Tokenization	46
4.4.2	Language Identification	47
4.4.3	Part of Speech Tagging.....	49
4.4.4	Normalization	51
4.4.5	Stopword Removal.....	54
4.4.6	Morphological Analysis.....	56

4.4.7	Index Term Selection	61
4.5	Translation.....	62
4.5.1	Text Translation	63
4.6	Concept Mapping.....	65
4.6.1	Lexical Mapping	68
4.6.2	Semantical Mapping	69
4.7	Classification.....	73
4.7.1	Text Categorization.....	73
4.8	Prototype	75
4.9	Summary	77
Experiment	78
5.1	Introduction.....	78
5.2	Data Collection.....	78
5.3	Implementation.....	81
5.4	Evaluation.....	81
5.4.1	Evaluation Metrics	81
5.4.2	Test Result	82
5.4.3	Discussion.....	88
Conclusion and Recommendations	90
6.1	Conclusion	90
6.2	Contribution of the study	91
6.3	Recommendations.....	92
References	93

List of Figures

Figure 2.1:Pre-processing steps in automatic text categorization.....	14
Figure 2.2: Example of a small ontology.....	22
Figure 2.3: A taxonomy of ontologies based on their purpose.....	23
Figure 4.1: Design of Multilingual Classifier for Under-resourced Language Documents	45
Figure 4.2: Tokenization.....	46
Figure 4.3: Language Identification.....	47
Figure 4.4: Part of speech tagging	49
Figure 4.5: POS label conversation	50
Figure 4.6: Homophone character normalization	52
Figure 4.7: Word Expander.....	53
Figure 4.8: Stopword Removal.....	55
Figure 4.9: HornMorpho combiner.....	61
Figure 4.10: Index Term Selection	62
Figure 4.11: Text Translation	63
Figure 4.12: Translation of Bag of tagged objects to equivalent sense of target language	64
Figure 4.13: Hierarchical structure of Newscodes.....	66
Figure 4.15: Concept Mapping	68
Figure 4.16: Sense to ontology concept mapping.....	72
Figure 4.17: Text Categorization	74
Figure 4.18: The screenshot of prototype's user	75

List of Tables

Table 2.1: Example of machine translation: lexical variety and ambiguity	27
Table 4.1: Sample of treetagger output - English parameter file	38
Table 4.2: POS label configuration for Amharic text language.....	51
Table 4.3: Sample of Amharic and Tigrinya Normalized.....	52
Table 5.1: Treetagger training corpus size statistics.....	79
Table 5.2: Bilingual dictionary size statistics	80
Table 5.3: Confusion matrix for experiment 1.....	83
Table 5.4: Precision, Recall and F-measure results for experiment 1	83
Table 5.5: Confusion matrix for experiment 2.....	84
Table 5.6: Precision, Recall and F-measure results for experiment 2	85
Table 5.7: Confusion matrix for experiment 3.....	86
Table 5.8: Precision, Recall and F-measure results for experiment 3	86
Table 5.9: Confusion matrix for experiment 4.....	87
Table 5.10: Precision, Recall and F-measure results for experiment 4	88

List of Algorithms

Algorithm 4.1: Tokenization.....	47
Algorithm 4.2: Language identification.....	48
Algorithm 4.3: POS label conversation	50
Algorithm 4.4: Homophone Characters Normalization.....	53
Algorithm 4.5: Word Expander	54
Algorithm 4.6: Stopword Removal.....	56
Algorithm 4.7: HornMorpho analyzed output	57
Algorithm 4.8: Lemma Information per word indexing	58
Algorithm 4.9: Lemma extraction for translation	59
Algorithm 4.10: Word lemma selection for index term selection	60

Lists of Acronyms

CPC	Category Pivoted Categorization
DAML	DARPA Agent Markup Language
DPC	Document Pivoted Categorization
FS	Feature Selection
IDF	Inverse Document Frequency
IPTC	International Press Telecommunications Council
IR	Information Retrieval
KNN	K-Nearest Neighbors
LSA	Latent Semantic Analysis
ML	Machine Learning
NLP	Natural Language Processing
OIL	Ontology Inference Layer
OOP	Object Oriented Programming
POS	Part Of Speech
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SVM	Support Vector Machines
TF	Term Frequency
TC	Text Classification
W3C	World Wide Web Consortium
WNO	World News ontology
XML	eXtensible Markup Language

Chapter One

Introduction

1.1 Overview

With the rapid growth of the Internet, digital text documents are increasingly replacing the printed ones. Today, searching books and news electronically is becoming the most popular way for capturing document and information.

Almost all companies have web pages and share their information on Internet. This “deluge” of documents originates needs for their automatic classification in order to accelerate the search of specific information. Organizing a large amount of documents manually is extremely expensive, time consuming, difficult and, is often impossible to do. Automated text categorization could help to do this hard task.

Automated text categorization, a subfield of NLP, aims at classifying documents to one or more categories [1]. A category is represented by a label, and may refer to a class or concept.

Recently automatic text classification of digitized documents gained a higher significance due to the rapid growth of digital content. With respect to the growth, organizing them is a big challenge for efficient retrieval of relevant information. Therefore, finding and improving solutions for text classification has considerable importance. In addition, it is extensively used in a wide and diverse range of practical works like spam filtering [2], electronic news classification [3], email classification [4], web page classification [5], and many others.

Many works of automatic text categorization is manipulated based on representative keywords or concepts [8]. In keyword based approach keywords are extracted from the document to identify the category of a given document. In this approach a document that is going to be categorized should contain a specific keyword that matches the represented document to be categorized into the predefined class. Due to this, keyword based approach can be spoiled due to some existing practical constraints. Out of the number of limitations, existence of vocabulary ambiguities in natural languages makes the situation worse. As the classification schemes are based on natural languages, these ambiguities are inherited in them too and this causes to reduce the accuracy of the classification results. Therefore, the incidence of Ambiguous terms in natural language may cause to reduce the accuracy of the classification process. Types of Ambiguous terms in natural

languages appear in numerous forms such as synonyms, homonyms and so on. Basically, the incidence of such types of vocabulary ambiguities poses major challenge in keyword based classification.

Instead of using keywords, documents can also be classified by taking into consideration the concept that the document represents. Concept is a "semantic" or meaning of terms. Terms are words that describe concepts or act as synonyms for other concepts. Hence, concept based text categorization allows classification of documents based on meaning rather than keywords. This method extracts concepts from the document and uses those concepts to categorize the document [11].

In order to use concepts to categorize documents, the concepts should be represented in the knowledge base. Representing such concepts in the knowledge base is provided using ontologies. Ontologies are currently considered as the de-facto standard for representing semantic information and it is a systematic formalization of concepts, definitions, relationships, and rules that captures the semantic content of a domain in a machine-readable format [12].

As explained above, concept or ontology based text categorization is an approach that outperforming keyword based approach. However, it is difficult to adopt such advantageous text categorization approach for under-resource language like Amharic, Afaan Oromo and Tigrinya languages. Since, language resource is important in order to build ontology to use as a knowledge base of the document categorization and because of the resource scarceness of such under-resource languages.

In order to attain the ontology based text categorization for such under-resource languages without consideration of previously explained difficulty, an approach which can use ontology of resource rich languages as a knowledge base and can classify the documents written in under-resource language without language barrier is required.

On the other hand, in the world wide scenario of the web page, multilinguality is a crucial issue to deal with and to investigate, leading us to reformulate most of the classical NLP problems.

Moreover, during this last decade, research paid an important attention for treatment of multilingual data. Due to a de facto of such multilingualism environment, most applications of text classification which are mentioned earlier also turn out to be interesting application of multilingual text classification, where documents given in different languages are to be classified by topic or similar criteria.

In general the ontology based text classification without language barrier on a multilingual environment can be achieve in two high level approaches [18] :-

- **Approach by Translation** – this is an approach which handles classification of text document with multiple languages over an ontology of a single language using translation. However, the translation of text is a difficult task and is never perfect. To minimize this risk, most approach use translation of term vectors which are extracted as representative of the given document and not the whole text. This approach provides a modular and flexible framework for addressing such a multilingual text classification problem.
- **Approach by Multilingual Ontology:** - This solution bypasses machine translation in multilingual environments by using a single ontology system to which predetermined manually translated associated concepts in multiple language. To perform mapping of ontology concepts in to different language manually needs human expert and due to this it needs more time, more effort and also error prone.

Due to lack of available language resource of good quality and broad coverage of under-resource languages, especially Ethiopian languages such as Amharic, Afaan Oromo and Tigrinya, it is difficult to investigate a good performing concept based text categorization

So, this investigation aims to design and develop an approach which can classify documents written in under-resourced language using resourced language ontology (i.e. English ontology) as a knowledge base. Beside this, apart from truly multilingual environments as in Ethiopia, the proposed approach can classify textual documents written in Amharic, Afaan Oromo or Tigrinya with ontology driven approach without language barrier.

1.2 Statement of the Problem

Due to the exponential growth of information on the internet, the difficulty of finding and organizing this information disseminated in the four corners of the planet, and implies a de facto multilingualism. So, due to this reason techniques and tools are needed which operates beyond language barriers. Multilingual tools become more important with respect to continuing integration of society. Beside this to enable knowledge sharing and reuse, it is necessary to represent concepts and relations in multiple languages [8]. However, prior research has not paid much attention to solve this problem for documents written in under- resourced languages (i.e. Amharic, Afaan Oromo and Tigrinya). To our best knowledge in order to alleviate such problem researchers investigated for documents written in Amharic, Afaan Oromo and Tigrigna languages separately.

Only one research was conducted for Afaan Oromo text categorization, which was done by Kamal Mohammed Jimalo, Ramesh Babu P and Yaregal Assabie [112]. Similarly, one research was conducted for Tigrinya text categorization, which was done by Geberehiwot [13]. For both languages text categorization, the researchers used a machine learning approach and due to this still there are a number of issues that are not addressed yet, which needs further improvement. In addition to this, there was also a research work that had investigated to solve the problem of keyword based text classification through categorizing documents based on their concept for Amharic text documents only [8] and it achieved a promising result. But, due to a defacto of multilingualism a tool which operates beyond a language barrier is required.

The problem of classifying multilingual pieces of text was addressed since the end of the last millennium [14] but it is still a significant problem because each language has its own peculiar features, making the automatic management of multilingualism an open issue. Particularly, the research and development of ontology based text categorization in a multilingual environment is in its formative state. To our best knowledge only few researches were conducted in this area.

Furthermore, building a knowledge base for these under-resources language (i.e. Amharic, Afaan Oromo, and Tigrinya) is very challenging. Because building a knowledge base (i.e. Ontology) needs domain knowledge as well as language resource. However, due to lack of language resource of good quality and broad coverage, it is very challenging and difficult to investigate a

knowledge base text classifier. Hence, due to such challenge of building an ontology with under-resourced language, ontology based text classification is limited to resourced language (i.e. English) support.

The main issue of this investigation is aim to solve the problem of classifying textual documents written in under-resourced language using resourced language knowledge base. As well, this investigation also motivated to provide multilingual text classifier to organizations and individuals in the increasingly globalized and multilingual environment. Specifically, this investigation proposes a text categorizer which uses English ontology to classify documents written in Amharic, Afaan Oromo and Tigrinya.

Therefore, the researchers set the following research question to examine the problem in classification of under-resourced languages document using ontology based text classification approach.

1. How to classify under-resourced language documents on top of English ontology?
2. How and when the performance of the proposed multilingual under resourced language document classifier increases?
3. To what extent, additional features incorporated in this study affect the performance and quality of the proposed document classifier?
4. Which feature contributes more, and less, to the performance of the proposed document classifier?

1.3 Objectives

1.3.1 General Objective

The general objective of this research is to investigate categorization of under-resourced languages documents with resourced language ontology, particularly focusing on Amharic, Afaan Oromo and Tigrinya documents.

1.3.2 Specific Objectives

The following specific objectives are identified in order to achieve the specified general objective:

- Review literature on the concepts of text classification, in depth on a multilingual ontology based text classification.
- To collect dataset to train integrated components for ontology based multilingual text categorizer
- To design a generic model for ontology based multilingual text categorizer.
- To develop a prototype for ontology based multilingual text categorizer.
- To conduct experiments to evaluate the usability of the proposed system.
- Forward conclusion and recommendation based on the experimental results.

1.4 Methodology

To accomplish the objectives of the research, the following methodologies are followed

1.4.1 Literature Review

In order to design an effective concept based multilingual under-resourced language text categorizer, any available works related to automatic text categorization; particularly ontology based text categorization on multilingual environment for other languages are reviewed thoroughly.

1.4.2 Data Collection

To make the proposed text classifier more powerful on a multilingual environment, different corpuses for all supported language of classifier is required such as corpus for part of speech tagging, for bilingual dictionary, for language identification and so on.

1.4.3 Design the Framework

Automatic text classification task has various steps and modules that can be used at each stage in order to develop the classifier. Framework that can categorize multilingual under-resource language documents using-resourced language ontology is designed.

1.4.4 Prototype Development

In order to show the usability of this work, prototypes for multilingual under-resourced text classifier that can categorize multilingual under-resourced text based on their concept is developed using Java programming language. Since, Java is suitable for different NLP tasks and also it is object oriented programming; it supports reusability, inheritance and easy to expand.

1.4.5 Evaluation

The outcome of the study is evaluated with the appropriate evaluation techniques to verify that whether the goal is achieved or not. The proposed prototype is tested for correctness using unseen documents. The result, which is automatically classified documents, is checked against the manual classification. Beside this, the performance of the text classifier are measure through common evaluation metrics such as recall, precision and F-measure.

1.5 Scope and Limitation

The scope of the study is to propose a model and develop an automatic classifier for multilingual under resourced documents using concepts. To make conceptual or semantic categorizer for experimental purpose only, news ontology is used, which encompasses formulating news domain concepts, building relations between concepts and representing restrictions. Beside this, the classification process of proposed work cope only documents written in Amharic, Afaan Oromo and Tigrinya. In addition, it is only considering a text document that contains sequence of alphabets of those languages without any figure, table, images or pictorial representation.

1.6 Application of Results

Resourced language ontology based multilingual under-resourced text categorizer plays an important role in a wide variety of information management tasks; particularly for under-resourced languages and the findings of this work can be used:

- In search engines to improve search result.
- For filtering and categorizing news items for any interest group, especially for news agencies.
- To organize and improve browsing multilingual web documents.
- For any organization which has a large collection of multilingual documents to automatically categorize documents for better management, and
- Moreover, the result of the study is play a role in academics for future study in the area of text categorization; specifically ontology based multilingual text categorization

1.7 Thesis Organization

This thesis consists of six chapters organized as follows: Chapter 2 (Literature Review) focuses on the background and theoretical concepts related to the document classification, steps in automatic document classification. A classification of ontologies according to their level of details is presented and the use of ontologies for information integration is reviewed. The chapter also presents components of the ontology, ontology languages, tools and different methodologies for building ontologies. Chapter 3 (Related Work) reviews the related work in the domain of Automatic document classification at monolingual as well as multilingual environment, presents a reviews of related works based on an approach used i.e. keyword based and concept based automatic classification techniques. Architecture of multilingual under-resourced documents classifier based on resourced ontology. Chapter 4 (Design and Implementation of Proposed Document Classifier) present the steps of implementing the methodology. It describes the proposed approach, the construction of domain ontology, the process of documents annotation and category assignments. Presents the proposed framework of automatic ontology based multilingual document categorizer. It presents the design as well as the implementation issues of the proposed approach. Chapter 5 (Data Collection and Preparation) gives statistics information of collected corpus used for proposed approach with their source in which the data are collected. Chapter 6 (Experiment and Analysis) presents an evaluation of the proposed approach and discusses the results. The last chapter (Conclusion and Recommendation) concludes the thesis and presents future directions.

Chapter Two

Literature Review

2.1 Introduction

This chapter provides a brief description about basic concepts of text classification. In addition, it also presents about category of text classification based on different criteria (i.e. application, usage and automation) and the application of text classification. Beside this, it also deals about the general steps of automatic text classification such as pre-processing, classification and performance evaluation. Finally, it also introduces about the concept of ontology and issues related to multilingual based text classification.

2.2 Text Categorization

In recent years the large amount of electronic data made available from a variety of sources, which include unstructured and semi-structured information, raises the need to devise automatic methods in order to extrapolate information from these data. In this context, the text mining studies are gaining more and more importance.

The main goal of text mining is to enable users to extract information from textual resources and deals with operations like retrieval, classification and summarization. Natural Language Processing, Data Mining, and Machine Learning techniques work together to automatically classify and discover patterns from the different types of documents [38].

Text Classification is an important part of text mining and it is a task of classifying set of natural language documents into a predefined set of categories based on their content [52]. This assignment requires checking similarity of the documents for a given set of categories and it can be used for classification, filtering, and retrieval purpose. For instance we might want to classify each incoming news document with a topic like “sport”, “politics” or “art” based on the content in which the document talks about.

Text classification extensively used in a wide and diverse range of practical works, some examples of domains in which text classification commonly used are:-

- **Spam Filtering:** it is often desirable to classify email [67] [68] [31] in order to determine if an email is spam [53] or a legitimate one, in an automated way.
- **Email Routing:** routing an email sent to a general address to a specific address or mailbox depending on topic [46] (i.e., work, friend and so on).
- **Language Identification:** Automatically detecting the language(s) present in a document based on the content of the document.
- **Readability Assessment:** Automatically determining the degree of readability of a text, either to find suitable materials for different age groups or reader types or as part of a larger text simplification system.
- **Opinion Mining:** Customer reviews or opinions are often short text documents which can be mined to determine useful information from the review. In particular the Opinion Mining process (also known as Sentiment Analysis) consists in determining the attitude of a speaker or a writer with respect to some topics or the overall contextual polarity of a document.
- **News Filtering and Organization:** Most of the news services today are electronic in nature in which a large volume is created very single day by the organizations. In such cases, it is difficult to organize the news articles manually. Therefore, automated methods can be very useful for news categorization in a variety of web portals [47] [49]. Providing tools for ontology driven text classification in the context of news text classification is one of the goals of our thesis and will be discussed in details throughout this document.
- **Opinion Spam Filtering:** Automatically detecting deceptive opinions. Nowadays a large number of opinion reviews are posted on the Web. Such reviews are a very important source of information for customers and companies. The former rely more than ever on online reviews to make their purchase decisions, and the latter to respond promptly to their clients' expectations. Unfortunately, due to the business that is behind, there is an increasing number of deceptive opinions, that is, fictitious opinions that have been deliberately written to sound authentic, in order to deceive the consumers promoting a low quality product (positive deceptive opinions) or criticizing a potentially good quality one (negative deceptive opinions) [32]

Analyzing these contexts, the need for text classification becomes very clear; However categorizing and grouping documents manually by human experts can be extremely laborious and time consuming , not only this it also requires a certain level of vocabulary recognition and knowledge processing. But, this problem can be alleviated by means of automatic text categorization [66].

Automatic text categorization is the process of automatically classifying a set of documents into predefined categories. The automatic categorization process is a combination of information retrieval (IR) technology and knowledge representation technology. In general terms, it is a process of classifying a given document into one or more predefined classes [39].

Text categorization can be divided into different categories using different criteria [40]. Depending on the application area, text categorization can be single-label or multi-label, on the other hand depending on the use of a text classifier text categorization can be document-pivoted, or category pivoted, further more based on the automation of the system text categorization can be hard or soft.

2.2.1 Text Categorization based on Application Area

In text classification a category is represented by a label, and may refer to a class or concept. Depending on the application area, a text categorization problem could be categorized as a single label or a multilabel [41]. In a single label categorization problem assigns only one predefined category to each “unseen” natural language text document and often defined as non-overlapping [54]. In this label for a given integer k each element of C must be assigned to exactly k (or $< k$, or $> k$) elements of D . For instance, this happens when the category needs to be evenly populated [43]. So, it assigns an object to exactly one category when there are two or more categories in the category spaces. However, it is impossible to categorize each document under a single label because of the nature of the text overlapping each other in the category spaces. For example, the economics field often overlaps (relates) with the political science field. This fact forces the different constraints on single-label text categorization task.

Whereas the multi-label case is general case in which any number of categories from 0 to M (M is at least one) may be assigned to the same document [44]. The multi-label text categorization

assigns more than one predefined category to an “unseen” document and it is called as overlapping text categorization tasks because it is the task of assigning an object simultaneously to one or multiple category. In this types of text classification the forecast may be right , wrong or partially right; because in case documents belong to two or more classes , the projection can hit them (all classes), any of them or just a few of them. This is because a document may contain multiple concepts. Providing multilabel text classification in the context of news domain is one of the goals of our thesis and will be discussed in details through the document.

2.2.2 Text Categorization based on Usage

Depending on the usage a text categorization can be Category Pivoted Categorization (CPC) and Document Pivoted Categorization (DPC). The CPC fills the decision matrix one row at a time where as the DPC fills the decision matrix one column at a time. According to [50], DPC is commonly used when documents are available at different moments in time, for example in filtering e-mail. However, CPC is used mostly when a new category $c_{|c|+1}$ is needed to add to an existing set after a number of documents have already been classified under it ,and these documents need to be reconsidered for classification under $c_{|c|+1}$.

2.2.3 Text Categorization Automation

A text categorization can be also soft and hard depending on the automation. The hard categorization completely automates the text categorization which requires a true or false decision for each pair (d_j, c_i) where soft categorization uses partial automation of the text categorization system which requires different methods. For a given document d_j in a documents D , the system may rank the categories in $C = \{c_1, c_2... c_{|C|}\}$ according to their estimated appropriateness to d_j without taking any hard decision on any of the categories. This is useful especially in critical applications in which the effectiveness of a fully automated system may be expected to be significantly lower than that of a human expert. This ranked list would have a great advantage for human expert for taking the final decision, because she/he would rank the categories based on his /her choice [21].

For automatic categorization text it is preferable to use hard categorization because the hard categorization fully automates the text documents of the specified language [40].

Beside these, there are also two main categories of text classification approaches. These are flat text classification and hierarchical text classification [21]. In flat text classification, categories are treated in isolation of each other and there is no structure defining the relationships among them. A single huge classifier is trained which categorizes each new document as belonging to one of the possible basic classes. In general in flat text categorization, the single label is the commonly used mechanisms of text classification. Such simple approaches work well on small problems, but they are likely to be difficult to use when there are a large number of classes and features.

For such large problems, hierarchical approach is suitable; the classification problem can be decomposed into a set of smaller problems corresponding to hierarchical split in the tree [62]. In general in hierarchical text classification, the multi-label text classification mechanisms are used commonly. For example, in a document for sport the main category called sport may have sub-categories under it like athletics, football, basketball, ground tennis, etc. In our study to enhance the performance of the text classification a hierarchical text classification approach instead of flat text classification approach is adopted.

2.3 Steps in Automatic Text Classification

With the aim of categorizing a given document into predefined categories, automatic text classification involves three main steps: pre-processing, classification and performance evaluation steps.

2.3.1 Pre-processing

In order to perform automatic text classification, the document must first be prepared to an acceptable representation that can be used by the next classification step. This preparation is represented by a great amount of features and this will then bring major benefit of data size reduction which increases performance in terms of memory size and processing time. Usually in this phase there are standard actions performed on the document [42]:

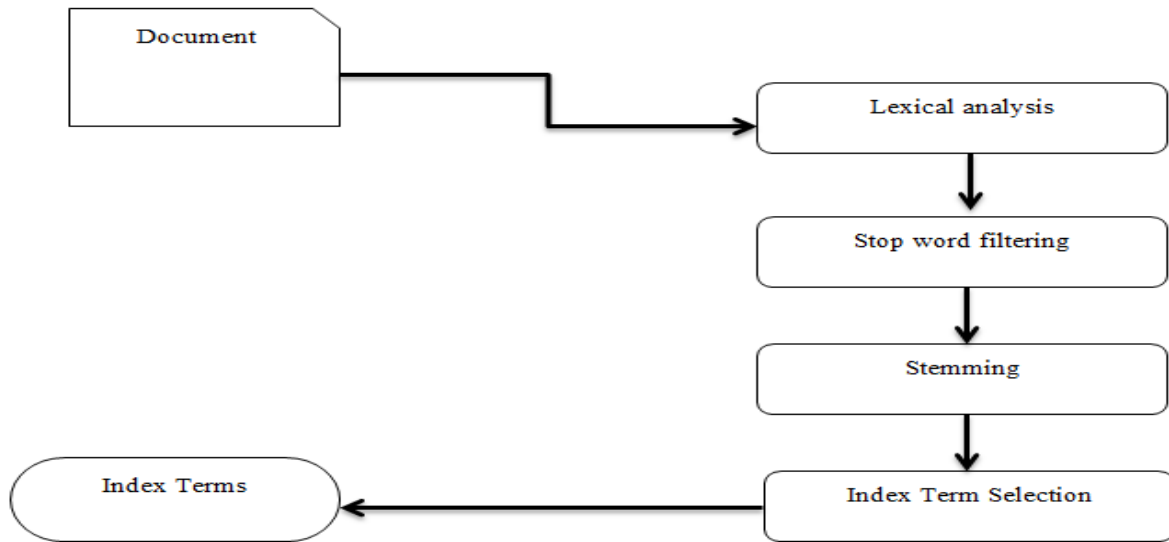


Figure 2.1:Pre-processing steps in automatic text categorization

(I). Lexical Analysis

Lexical analysis is the first task in the lexical analysis is tokenization, which is the process of converting the stream of characters in a document into tokens or list of terms, where a term is defined as a string of letters, digits or other special characters, separated by punctuation marks and spaces. In addition to tokenization, lexical analysis includes data cleaning , which is the process of removing characters like punctuation marks , special symbols and so on from the given text document. Since such types of characters are not important for the purpose of classification and should be removed.

(II). Stop word Filtering

Stop-words are words that occur frequently in the document, but have no impact to discriminate among documents. Examples of English stop words: “the”, “a”, “and”, etc. Such frequently used words generally “glue” sentences together but they usually do not carry meanings. Such types of words are not important for the purpose of text classification, so from the point of text classification, applying stop-word removal reduces the complexity of the document representation and the number of tokens to be processed.

(III). Stemming

Stem words are words that appear in a document often have many morphological variations. In most cases morphological variations have similar interpretation and can be considered as equivalent for the purpose of IR applications. Therefore, in stemming different forms of the same word are consolidated into a single word. Thus, terms of a document are represented by stem

words rather than by the original words. For example, singular, plural and different tenses are consolidated into a single word. This also reduces the number of different terms needed for representing a document and also saves storage space and processing time. There are a number of stemming algorithms, such as table lookup approach, successor variety, ngram stemmers and affix removal.

(IV). Indexing and Features Selection

The main idea is to reduce the complexity of the documents and make them easier to handle; the document has to be transformed from the full text version to a document vector and, after that, it is performed the Feature Selection (FS) [22] that is the selection of subset of features from the original documents. FS is performed by keeping the words with highest score according to predetermined measure of the importance of the word. There are different techniques of selection of index terms to represent a document. Among the most widely used techniques to select document representative terms are statistical techniques. Statistical techniques for text analysis are based on Term Frequency (TF) and Inverted Document Frequency (IDF).

TF also known as “bag of words”, in which a document is represented as a set of words, together with their associated frequency in the document. Such representation is essentially independent of the sequence of words in the collection. The frequency of words in the document determines which words were sufficiently significant to represent the document. It is based on the principle that a term which is frequently used in a document is useful to represent the document. In short, the frequency shows usefulness of the term in the document, and it is formally stated as follows [69]:

$$T_f(d_j, t_k) = \sum_{j=1}^n f(T_j) \quad (2.1)$$

TF (d, t), is the number of times a term t occurs in the document d and is defined as: Where d_i is the i^{th} document, t_k is the k^{th} term of document d_i and T_j is the j^{th} term in the document. Most text classification methods use the bag of words representation because of its simplicity for classification purposes.

IDF is the occurrence of a term in the collection of all input documents; if a word occurs in all documents, the relevance of the document will decrease because the probability of the word to represent the document is less. That is, terms that appear in many documents are not very useful as they do not allow discriminating between documents. A formal definition of IDF is: [42].

$$idf(t) = \log\left(\frac{N}{df(t)}\right) \quad (2.2)$$

Where $df(t)$ is the number of documents including the term t , N is the number of all documents. It is possible to use either term frequency or inverted term frequency depending on the application.

However, term frequency is more appropriate for this study because the study considers a single document at a time instead of collection of documents to categorize.

2.3.2 Classification

This is the real heart of the TC task. The document prepared in the previous steps can be classified by two principal approaches as described below.

- **Machine learning approach:** the most common used approach; it is based on standard ML techniques in order to classify a text with respect to a set of documents previously labelled (training set).
- **Knowledge based approach:** it is mainly based on NLP techniques in order to classify a document using the semantic knowledge like, for example, the semantic relationships among the words (e.g., synonym, antonym, etc.). Normally this approach uses ontologies to represent the knowledge model, as presented in our work.

A. Machine Learning Approach

A Machine Learning text classification task starts with a training set $D = (d_1 \dots d_n)$ of documents that are already labelled with a class identifier C (e.g. sport, politics); the task is then to determine a classification model which is able to assign the correct class to a new document d . Moreover the classification using this approach can be supervised; semi supervised and unsupervised learning [23].

The machine learning approach in the area of text classification is so vast that is impossible to cover all the different algorithms in detail in a single topic here. Therefore, our goal is to provide

an overview of the pointers to the different variations of this technique most important supervised methods, and also the.

In supervised Machine Learning methods, a model is created based on previous observations, i.e., a training set. In the case of document classification, categories are predefined and a training dataset of documents is manually tagged as part of a category. Following the creation of a training dataset, a classifier is trained on the manually tagged dataset. The idea behind this approach is that, the classifier will then be able to predict any given document's category from then on. Therefore, designing classification methods that effectively account for these characteristics of text is of paramount importance. Some widely adopted methods, which are commonly used for text classification:

- **Decision Trees:** - Decision trees are designed with the use of a hierarchical division of the underlying data space with the use of different text features [55]. The hierarchical division of the data space is designed in order to create class partitions which are more skewed in terms of their class distribution. For a given text instance, we determine the partition that it is most likely to belong to, and use it for the purposes of classification.
- **Pattern (Rule)-based Classifiers:** - In rule-based classifiers we determine the word patterns which are most likely to be related to the different classes. We construct a set of rules, in which the left-hand side corresponds to a word pattern, and the right-hand side corresponds to a class label. These rules are used for the purposes of classification [56].
- **SVM Classifiers:** Support Vector Machines (SVM) Classifiers attempt to partition the data space with the use of linear or non-linear delineations between the different classes. The key in such classifiers is to determine the optimal boundaries between the different classes and use them for the purpose of classification [64].
- **Neural Network Classifiers:** - Neural networks are used in a wide variety of domains for the purposes of classification. In the context of text data, the main difference for neural network classifiers is to adapt these classifiers with the use of word features. We note that neural network classifiers are related to SVM classifiers; indeed, they both are in the category of discriminative classifiers, which are in contrast with the generative classifiers [24].

- **Bayesian (Generative) Classifiers:** - In Bayesian classifiers (also called generative classifiers), we attempt to build a probabilistic classifier based on modeling the underlying word features in different classes. The idea is then to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word presence in the documents.
- **Other Classifiers:** - Almost all classifiers can be adapted to the case of text data. Some of the other classifiers include nearest neighbor classifiers [25], and genetic algorithm-based classifiers.

In order to use this supervised ML algorithm, we need for a training set; however, it is often the case that a suitable set of well categorized (typically by humans) training documents is not available. Even if one is available, the set may be too small, or a significant portion of the documents in the training set may not have been classified properly. The classifier performance depends heavily on the large amount of hand-labeled documents as they are, the only source of knowledge for learning the classifier. Being a labor-intensive and time consuming activity, the manual attribution of documents to categories is extremely costly. To overcome these difficulties, semi-supervised learning techniques have been proposed that require only a small set of labeled data for each category [63].

The problem is that all of these methods require a training set of pre-classified documents and it is often the case that a suitable set of well categorized (typically by humans) training documents is not available. This creates a serious limitation for the usefulness of the above learning techniques in operational scenarios ranging from the management of web-documents to the classification of incoming news into categories, such as business, sport, politics, etc.

Most important, text categorization should be based on the knowledge that can be extracted from the text content rather than on a set of documents where a text could be attributed to one or another category, depending on the subjective judgment of a human classifier.

B. Knowledge Based Text Classification

This text categorization method is based only on leveraging the existing knowledge represented in domain ontology. The positive aspect of this approach is that it is not dependent on the existence of a training set, as it relies solely on the entities, their relationships, and the taxonomy of categories represented [34], for example, in an ontology, that effectively becomes the classifier. So to face the issues about the lack and the subjectivity of manually labelled datasets, the basic idea is to use a knowledge based approach. Since, training with a set of pre-classified documents is not needed, as the ontology already includes all important facts. So the knowledge represented in such a comprehensive ontology can be used to identify topics (concepts) in a text document, provided the document thematically belongs to the domain represented in the ontology. Furthermore, if the concepts in the ontology are organized into hierarchies of higher-level categories, it should be possible to identify the category (or more categories) that best classify the content of the document.

As an example of ontology text classification, let us assume that we have a well-defined and comprehensive ontology containing knowledge about the smartphone domain. The ontology includes a wide variety of concepts about smartphone features, such as brands, display, and type of connection technology and so on, organized into a hierarchy structure. Now, let us consider an article, maybe a review, describing a new smartphone:

*"This **phone** is just perfect! Good **screen** and **battery life**, the **front camera** is 3.0 mpx and **design** is simply perfect. It supports also **4g connection**!"*

Within this document we will be able to identify a large number of concepts present in our ontology (the bold words). As mentioned previously, this approach does not classify the document with respect to a set of classes, but it is able to classify it in respect of those categories represented by the ontology. One shortcoming of this approach is the presence of false positives; in fact we could classify texts in a wrong way because of the presence, inside the ontology, of concepts less specific to its domain, (i.e., they can be present in more contexts).

In this investigation the ontological knowledge on news domain is used, however this knowledge is not only used as lexical support, but also for deriving the final categorization of documents into news categories.

2.3.3 Performance Evaluations

This is the last step of TC, in which the evaluation of text classifiers is typically conducted experimentally, rather than analytically. The experimental evaluation of classifiers, rather than concentrating on issues of efficiency, usually tries to evaluate the effectiveness of a classifier, i.e. its capability of taking the right categorization decisions. Many measures have been used to determine the effectiveness' or the performance of the algorithm, the metrics like precision and recall and F-measure [70] are most often used.

Precision is determined as the conditional probability that a random document d is classified under c_i , or what would be deemed the correct category [40].

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.3)$$

Recall is defined as the probability that, if a random document (dx) should be classified under category (c_i), this decision is taken [40].

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.4)$$

Where

True Positive (TP) - situation in text classification when the classifier correctly classifies a positive test case into the positive class;

True Negative (TN) – situation in text classification when the classifier correctly classifies a negative test case into the negative class;

False Positive (FP) – situation in text classification when the classifier incorrectly classifies a negative test case into the positive class;

False Negative (FN) – situation in text classification when the classifier incorrectly classifies a positive test case into the negative class;

Precision and recall are often combined in order to get a better picture of the performance of the classifier given as F-Measure [28].

$$F - \text{Measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (2.5)$$

In this investigation, in order to measure the effectiveness of the proposed under-resource textual documents classifier over the multilingual environment using the resourced language ontology the above mentioned evaluation techniques (Recall, Precision and F-Measure) are adopted.

2.4 Ontology

Ontology is defined as “an explicit specification of a conceptualization” [65]. It can be viewed as a declarative model of a domain that defines and represents the concepts existing in that domain, their attributes and relationships between them [71]. Collectively, the concepts and the relationships form a foundation for reasoning about the domain. It is typically represented as a knowledge base which then becomes available to applications that need to use and/or share the knowledge of a domain [71], which the news domain in our thesis.

Ontologies specify a set of constraints that declare what should necessarily hold in any possible world. It used to identify what “is” or “can be” in the world. It is the intention to build a complete world model for describing the semantics of information exchange [35].

In the context of computer and information sciences, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically comprised of classes (or sets), attributes, and relationships (or relations among class members and constraints). In particular, classes are organized into hierarchies and define the types of attributes common to individual objects within the class. Moreover, classes are interconnected by relationships, indicating their semantic interdependence. Class hierarchies and class relationships form the schema level of the ontology, while the individuals (object instances or just instances) and links among them (relationship instances) form the so called ground level of the ontology.

Figure 2.3 represents a simple ontology also called lightweight ontology containing classes and its taxonomical relations

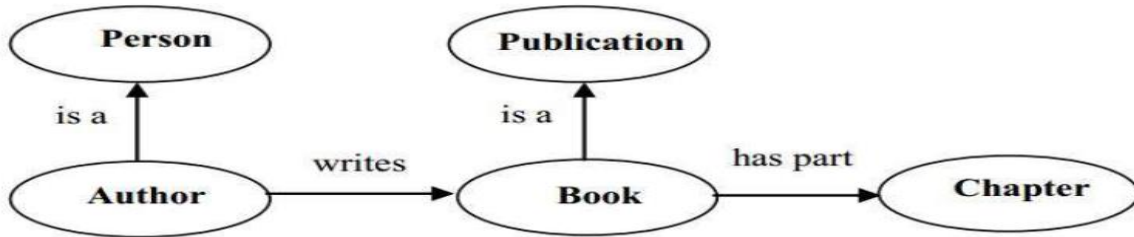


Figure 2.2: Example of a small ontology

There are several reasons to develop an ontology, related to the importance of having a common vocabulary to share information in a specific domain using a unique know “language”. Some of these reasons are:

- To share a common knowledge about information structure, among people and agents.
- To reuse existing domain knowledge in order to introduce standard languages and services for boosting interoperability.
- To provide reasoning services on a domain for knowledge evolution and understanding.

For example, let us consider several web sites that provide news information; if these sites share the same domain represented by unique news ontology of terms, the computer agents can extract and elaborate information from these sites in a standard way because they speak the same “language”. Often, developing ontology of the domain is not a goal in itself. Developing ontology is akin to defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data. Therefore, it is possible to identify different types of ontologies according to their application context (Figure 2.2):

- **Top - level or Upper or Foundational ontologies:** ontologies which describe very general concepts that are the same across all domains, which is independent of a particular problem or domain like space, time, matter and so on.
- **Domain Ontologies:** - contain the vocabulary of a specific domain (i.e. medicine, physics).
- **Task Ontologies:** - ontologies that formally specify the terminology associated with the types of task or activity (i.e. scheduling, selling). They allow terms to solve problems associated with a task that may or may not belong to the same domain.

- **Application Ontologies:** - are dependent on a particular application which contains all terms, concepts and relations that are needed to model a particular application under consideration. Usually a specialization of both Domain and Task ontologies.

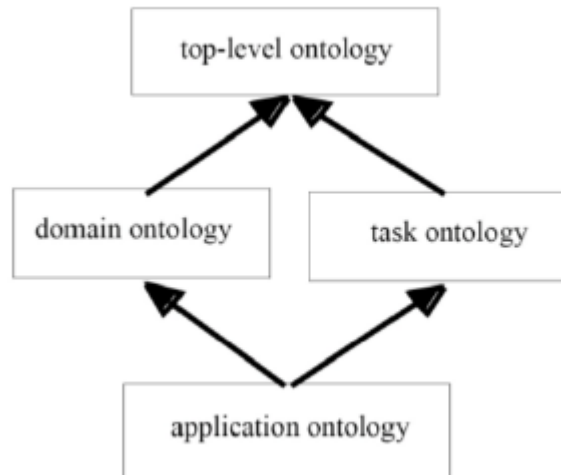


Figure 2.3: A taxonomy of ontologies based on their purpose

In natural language texts, the meaning of a term is usually not defined explicitly, but strongly depends on the context in which the term occurs. Humans are able to disambiguate them using their knowledge about the context the term is used in. Current automatic disambiguation approaches fail frequently due to missing commonsense knowledge or appropriate ontology models [37].

The advantages of an ontology-based classification approach over the existing ones are that the nature of the relational structure of ontology provides a mechanism to enable machine reasoning; also, the conceptual instances within ontology are not only a bag of keywords but have inherent semantics and a close relationship with the class representatives of the classification schemes [45].

2.4.1 Ontology Building

The creation of ontologies presents a tedious task, because it requires specialized skills and involves various stakeholders with personalized, depends on a variety of factors (such as software building tool, the implementation language, the development methodology, the

applications in which the ontology will be used, the type of the ontology under construction, the available informal and formal existing knowledge resources, etc.) [51].

In addition, there is no single correct way to model a domain, there are always viable alternatives. Therefore, there is no one correct way to develop ontology [48], but the quality of the solution depends on the skills of the people who will participate in the ontology development process. Several research groups have proposed various methodologies for building ontologies. The skilled knowledge engineer can look up the different methodologies before selecting, or adapting one that fits his needs [51].

Several methodologies for ontology building have been reported, includes *Cyc*, *Uschold and King's* method, *KACTUS*, *Methontology*, *SENSUS*, *On-to-Knowledge*, *Grüninger and Fox*, *TOVE*, *CommonKADS*, *DILIGENT* [72]. The most complete ones are *Methontology* and *On-to-Knowledge*. All these methodologies are composed of several activities. The development process is not a linear process but a refinement one where each activity can be repeated several times. Among all the activities the most important are: Ontology specification, Knowledge acquisition, Conceptualization, Formalization, Implementation, Evaluation, Maintenance, and Documentation [72].

2.4.2 Ontology Building Tools

There are a lot of software tools which are aimed at providing support for the ontology development process. Many ontology editors could be found on internet. Some of them (like: *Apollo*, *OntoStudio*, *Protégé*, *Swoop*, and *TopBraid*). All these tools are popular in the ontology design and development sector. They are accepted by relatively large semantic web communities [36].

These tools can be applied to several stages of the ontology life cycle including the creation, implementation, and maintenance of ontologies. It's used for building a new ontology either from scratch or by reusing existing one, which usually supports editing, browsing, documentation, export and import from different formats, and they may have attached inference engines [48].

Although, there are different ontology editors that can manage ontologies, the one used in this research is Portage, which provides a flexible plug-in architecture and plenty of different functionalities.

2.4.3 Ontology Building Languages

There are several languages used in ontology building like XML, RDF/RDFS DAML (DARPA Agent Markup Language) + OIL (Ontology Inference Layer) and OWL (Web Ontology Language). Many ontology tools have been developed for implementing metadata of ontology using these languages.

XML provides syntax for structured documents, without semantic meaning constraints on the documents. RDF is a data model for representing objects and relations between them. It provides simple semantics for the model and can be represented in XML syntax. RDF-Schema is a language for defining vocabulary for describing properties and classes of RDF resources. RDFS is used to define graphs of trio RDF, with semantics of generalization/prioritization of such properties and classes. OWL adds vocabulary for describing properties and classes, relations between classes (e.g. disjointness), cardinality and characteristics of properties (e.g. symmetry). OWL is developed as an extension of RDF vocabularies, and it is derived from the ontology DAML + OIL [36].

2.4.4 Ontology Evaluation

In order to build high quality ontologies, ontology evaluation technologies are needed. The primary goal of these evaluation methods is to prevent applications from using inconsistent or incorrect ontologies [29]. A variety of researches of ontology evaluation have been established depends on the perspective of what should be evaluated. Most of them focus on the evaluation of the whole ontology; others focus on partial evaluation of the ontology, for reuse it in an ontology engineering task [26]. Basically, ontology evaluation can be divided according to the following basis:

Corpus-based evaluation: is used to estimate empirically the accuracy and the coverage of the ontology.

Gold-Standard-based evaluation: that compares candidate ontologies to gold-standard ontology that serves as a reference.

Task-based evaluation: looks at how the results of the ontology-based application are affected by the use of ontology.

Expert-based evaluation: where ontologies are presented to human experts who have to judge in how far the developed ontology is correct.

Criteria-based evaluation: measures in how far ontology adheres to desirable criteria [26].

There are various methodologies to evaluate ontologies; most of them based on one of the following categories:

- Fitting or coverage techniques between ontology and a domain of knowledge that the ontology is created for.
- The effort done by human experts who try to assess how well the ontology meets a set of predefined criteria, standards, and requirements.
- Using the ontology in the context of an application or project to evaluate its effectiveness. The use of the system may reveal weakness or strength points in the ontology.
- Comparing the ontology with other ontologies in the same domain.
- Studying ontology relationships considering some measures.
- Studying and comparing the formal representation of the ontology with other ontologies formal representations, criterions, or measures [27].

2.5 Text Classification and Multilinguality

The first solution that comes to mind when working with multiple languages is to use translation. However, translation entails language-specific difficulties, such as the importance of the connection between grammar and meaning, the role of word endings and word position, and the length and complexity of words, which are comprised of other words. Translation also entails difficulties that arise from the translation effort itself: some words do not have exact parallels in other languages, nuances are hard to convey, and a word may have different meanings in different contexts.

Let us consider multilingual text classification (i.e., documents in multiple languages). If we want to classify a new document in a different language, in the case of supervised methods we

need a new training set for that specific language, and so on for each new document in different language that we want to classify.

Language	English machine Translation
Amharic	ሰው → Person
Tigrigna	ሰብ → Human

Table 2.1: Example of machine translation: lexical variety and ambiguity

Several solutions have been proposed to overcome this problem. We might think, for example, to translate documents in the language L of the training set. This approach has a number of shortcomings, for example:

- Lexical variety in L (e.g., English: huge vocabulary, many synonyms).
- Variety of expression in source language.
- Lexical ambiguity in L (unnecessary introduction of additional ambiguity).

These three points are well represented in the following example (Table 2.1). The word “ሰው” could be translated to “person”, while “ሰብ” into “human”. These different translations, depending on the language, get variety and ambiguity in the classification; in our work we are tries to handle these multilingual problems using ontologies.

In this investigation, in order to achieve a classifier, which classify multilingual under-resourced documents using resourced language ontology without language barrier a bilingual dictionary based word by word translation technique is adopted.

2.6 Under-resourced languages

Under-resourced languages are languages which have not large amount of language specific resources to solve the problem of most NLP related tasks. Since, most NLP related tasks including text classification needs large amount of training corpus that are not available for under-resourced languages. As one of the most under-resourced languages used in part of Horn of Africa which are official status at national and regional level, particularly in Ethiopia are Afaan Oromo, Amharic and Tigrinya.

Afaan Oromo, also called Afaan Oromo and Oromiffa, is a member of the Cushitic branch of the Afro-Asiatic language family. It is a macro language of Ethiopia, Kenya, Somalia, Eritrea, and Djibouti, by close to 40 million people, making it Africa's the fourth most widely spoken language after Hausa, Arabic, and Swahili. It is the statutory provincial working language in the Oromia Region of Ethiopia, one of the nine ethnically based regions of Ethiopia. It is used as a lingua franca by some 25.5 million people (Ethnologue) [11].

On the other hand, Amharic and Tigrinya are a member of the Semitic branch of the Afro-Asiatic language family. They are distantly related to Arabic and Hebrew. According 2007 report Amharic has close to 22 million first-language speakers and 4 million second-language speakers worldwide, of which slightly over 21.6 million live in Ethiopia (Ethnologue) [11]. Similarly, Tigrinya is spoken by 4.3 million people in Ethiopia, 2.8 million of who are monolingual speakers of the language. It is the third most commonly spoken language in Ethiopia where it serves as a lingua franca among the country's different ethnic groups. Population total of all countries who are speaking Tigrinya language is estimated at 6.9 million (Ethnologue) [11].

These three languages (i.e. Afaan Oromo, Amharic and Tigrinya) are used in the mass media, education, and in governmental and non-governmental agencies. In addition, large collection of these languages documents available in web, in addition to hard copy document in library, and documentation centers. Even though the amount of the document increase, there are challenge tasks in identify the relevant documents related to a specific topic [12]. So, a text categorization mechanism is required for finding, filtering and managing the rapid growth of online information.

2.7 Summary

This chapter explained what text categorization is as one of the IR application and dealing with classifying a set of natural language documents into a predefined set of categories based on their content. Beside this, it also indicates the wide and diverse range of application domain of text classification.

Through reviewing the different literatures it also explained the types of text categorization based on different criteria such as based on application (single label or multi label), based on usage (document pivoted or category pivoted) , and based on automation (hard or soft).

As well, in this chapter we were also discussed about steps in automatic text categorization such as preprocessing, classification and performance evaluation and each of these steps were explained briefly. Additionally, we also briefly discussed about ontology, ontology based text categorization and important activities, tools, languages, evaluation techniques to build ontology. Finally, we were discussed about multilinguality issue on a text categorization and under-resourced languages.

Chapter Three

Related Work

3.1 Introduction

In this section, we review the existing related work in the state-of-the-art on text classification in general, and multilingual text categorization in particular, and also introduce and analyze the related work in the field of ontology driven text categorization, aspects that characterize the contribution of our thesis.

3.2 Machine Learning Approach to Text Classification

A wide range of statistical and machine learning techniques has been applied to text categorization; in this section the most relevant works of text classification based on machine learning approach, particularly for local languages has been reviewed.

In the past years, few researches have been done in text classification for Amharic documents based on machine learning approach. However in this section, we only review the recent work done by [74]. The researcher attempted a hierarchical classification of Amharic news items using support vector machines. The research had been conducted with the aim of constructing hierarchical classifier and the experiment had been done using a categorical data collected from Ethiopian News Agency (ENA) to evaluate the performance of the hierarchical classifier over the flat classifier. The findings of the experiment show the accuracy of flat classification decreases as the number of classes and documents (features) increase, particularly when the number of top feature set increases. The peak accuracy of the flat classifier was 68.84% when the top 3 features were used. On the other hand, using hierarchical classification show an increasing performance of the classifiers as move down the hierarchy and the maximum accuracy achieved was 90.3% at level -3 (last level) of the category tree, particularly the accuracy increases when the number of top feature set increases as opposed to the flat classifier. The peak accuracy was 89.06% using level three classifier when the top 15 features were used.

Besides, the performance of flat classifier and hierarchical classifiers were compared using the same test data. Thus, it shows that use of the hierarchical structure during classification had

resulted in a significant improvement of 29.42 % in exact match precision when compared with a flat classifier.

As well, an automatic text classification for Afaan Oromo documents using machine learning technique, namely decision tree classifier and support vector machine had been investigated [112]. This study used annotated news texts to train these two classifiers with six news categories i.e. sport, business, politics, health, agriculture and education. In order to preprocess the Afaan Oromo documents, different text preprocessing such as tokenization, stemming, and stopword removal had been done. In order to conduct the experiment, 10 fold cross validation technique was used. The result of experiment indicated that Decision Tree classifier and Support Vector Machine on six news categories data achieved 96.8 % and 84.93 % respectively. As a result, the researchers concluded that, the machine learning classifiers were applicable to automatically classify the Afaan Oromo texts.

Beside this, an automatic text classification for Tigrinya text has been investigated by only one researcher [13]. This researcher introduced an automatic text classification for Tigrinya text documents with two step approach. In the first step, clustering was used to obtain natural group of the unlabeled data, specifically for this purpose the researcher used direct k-means and repeated bisection clustering algorithm and achieves 0.516 purity, 0.624 entropy and 0.56 purity, 0.611 entropy respectively. So, according to the result the researcher selected the repeated bisection clustering algorithm to train the text classifier. In the second step, classification was performed using Support Vector Machine (SVM) and j48 decision tree classifier. The SVM classifier correctly classified 82.4% with 32.68 seconds whereas the J48 classifier classifies 72% with 34.4 seconds. As a result, the researcher concluded that, the SVM classifier is effective and efficient in classifying the Tigrinya text documents.

3.3 Knowledge Based Approach to Text Classification

A wide range of works have been done in text classification based on knowledge based approach. In this section the most relevant works has been reviewed.

[8] Proposes a framework that automatically categorizes Amharic documents into predefined categories using knowledge represented in the News ontology. The document classification proposed by this paper has three stages. First, all the documents pass through pre-processing

stages. Then index terms are extracted from a given document which is mapped onto their corresponding concepts in the ontology. Finally, the selected document is classified into a predefined category, based on the weighted concept. The approach was tested and showed that the use of concepts for Amharic document categorizer results in 92.9% accuracy with a promising outcome.

[75] Propose an ontology based document classifier in order to improve the efficiency and effectiveness of Chinese web document classification and retrieval. The study constructs ontology based Chinese knowledge base named as HowNet and creates ontology for each subclass of the classification system. In this research RDFS was used in order to convert knowledge into ontology and to define the relations among ontology and an ontology relevance calculating algorithm can classify web documents automatically.

The approach was tested with SVM, KNN and LSA (TF-IDF) approaches for comparison purpose and according to the experiment result; SVM approach gets average precision rate of 80.1% and the average recall rate of 68.3%. KNN approach gets average precision rate of 82% and the average recall rate of 69.1% and the LSA (TF-IDF) approach gets average precision rate of 82.4% and the average recall rate of 73.8%. The proposed approach achieves experimental result of average precision rate of 81.9% and average recall rate of 75.8%. The experimental result showed that the proposed ontology based approach achieve highest average precision rate among other three methods (i.e. SVM, KNN, KNN and LSA) and its precision rate most stable.

[76] Focus on document classification based on the similarities of documents already categorized by ontology using terminology information from the documents. The document classification technique proposed by this paper does not involve any learning processes or experimental data and can be performed in real time. Their classification results, the precision, recall, and F1 measures are 89.68%, 95.43%, and 92.39% respectively. And the F1 measurement is compared with TF-IDF and Bayesian method which got 79.87% and 82.45%.

3.4 Text Classification and Multilinguality

In the past years, few researches have been done in text classification for multilingual environments. Some of the most relevant works are reviewed as follows:

T. Goncalves & P. Quaresma [20] proposed a method to combine different monolingual classifier in order to get a new classifier in which it was suitable for multilingual environment. To build the monolingual classifier, a supervised machine learning approach, particularly Support Vector Machine algorithm was used with labelled documents as training data set. The proposed method was applied to a corpus of legal documents in four different languages (i.e. English, German, Italian and Portuguese) and was evaluated.

For testing the proposed method, experiments were run over a set of European Union law documents, a set of 2714 full text legal documents. The experiments were done using a bag of words representation of documents over SVM algorithm for each language profile and evaluated using a 10 fold stratified cross validation procedure with significance tests done with 90% confidence level. To support the research claim, the experiment was conducted for each monolingual classifier and for all possible combiners. According to experimental result, when the Portuguese combiner combines with other language classifier and achieved an average precision of 0.831, when the English and German classifiers are combined resulted in with best average recall of 0.652 and average F1 measure of 0.709. Significance tests show that, for all classes and all performance measures, there was no significance difference between the “best” monolingual classifier and the corresponding combined classifier.

A. Segev. & A. Gal [18] used a lightweight method, which was a design based on multilingual ontology, which means representing the ontological concepts in multiple languages. Therefore, they aim at conveying the local interpretation of ontological concept, thus to overcome the language barrier. In order to analyze the impact of the proposed approach for the support of multilinguality data from news RSS (i.e. which is a format for distributing and gathering content from different sources in different languages across the web) a total of 1,778 data items in actual eGovernment environment had been used for experiment.

In the first experiment, the impact of multilinguality on a single class classification was evaluated in order to analyze the impact of multilinguality on classification recall and on average the use of multilinguality corpus results in a minor reduction of less than 2% in recall (from about 98.3% to 96.58%). On the other hand, in order to analyze the impact of

multilinguality on classification precision, it was evaluated on multi class classification. According to experimental result, the precision for all concepts reached 55.17% while the use of multilingual corpora reduced the precision by about 6% to 49.06%. These results indicate that the proposed approach suffers a minor reduction in performance with the introduction of multilinguality.

A.Ferrando et al. [19] used an approach by translation to handle classifying text documents without language barrier over an ontology built on top of a single language. In order to achieve this, the authors used the advantage of BableNet multilingual semantic network, in which words in different languages (specifically all European languages, most Asian languages, and even Latin) are grouped into sets of synonyms to cope with multilingual documents. The main goal of the thesis was to provide a modular and flexible framework for addressing such a multilingual text classification problem. In order to demonstrate the potential of the proposed approach the main field of text classification called sentiment analysis or opinion mining were implemented.

The experiment for the proposed approach had been conducted over a different threshold value (i.e. $Tr = 0$ and $Tr = 0.2$). In the first experiment when the threshold = 0 and in the first variant of SentiModule, smartphones reviews had been used as features ontology to represent the smartphone domain. The experiment results in concerning the reviews with evaluation 5 (i.e. very good) or 4 (good), in case of overall score 5 the approach correctly classifies only the 66.7% of the reviews and similarly only the 60.0% of the reviews with overall score 4. The worst classification results concern the reviews with evaluation 1 (i.e. very bad) and 2 (i.e. bad).

Indeed, respectively only in the 43.3% and 38.3% of the cases it produces a correct results. On the other hand, the results of the classification performed using Threshold = 0.2 and concerning the reviews with evaluation 5 (i.e. very good) or 1 (very bad) the approach correctly classifies the 92.6% and 89.1% of the reviews respectively. In case of overall score 4 (i.e. good) or 2 (bad) 88.8% and 82.6% of the reviews was correctly classified respectively.

3.5 Summary

In this chapter we presented a review of some related works in the field of text classification for various languages, including for Amharic, Afaan Oromo and Tigrinya local languages. The review showed that two major approaches can be followed for text categorization. The first approach is machine learning based approach for automatic text classification and the second is categorization through knowledge based approach. Even if all works reviewed based on machine learning approach solves the problem of manual classification of either Amharic, Afaan Oromo or Tigrinya documents, but there are still a number of issues that are not addressed yet. The primary issue is that to categorize documents in a certain category, it only considers selected keywords, which is not considered the core meaning or semantics of the document. In addition to this, it also influenced by word variations or ignores the semantic relationship between document's content and the designed category. Since the proposed work is ontology based text classifier, unlike machine learning based approach, it uses semantics of the document instead of keywords. Due to this, it considers the core concept of a document and also not influenced by vocabulary variation.

On the other hand, we reviewed the related works in knowledge based approach for text classification. As explained earlier, due to the important benefits of this approach, it makes a new and promising way of improving the categorization process. However, it is not able to operate beyond the language barrier or not support a de facto of multilinguality of information on the web and other sources. However, the proposed work is motivated to provide a multilingual text classification support, particularly for textual documents written in Amharic, Afaan Oromo and Tigrinya. Besides, the result obtained by T. Goncalves & P. Quaresma [20] was quite good. Since, the approach used by this research was a supervised machine learning approach i.e. combines a Support Vector Machine based monolingual classifier for different language and making suitable for multilingual environment. So, it has a number of issues which were not addressed such as ignorance of semantic between the document's content and designed category, influence of vocabulary ambiguities, and need of manual effort in labeling of training data. However, in the proposed work, there is no such problem since it is ontology based approach, training less and also not depends on the keywords rather on concepts or semantics of document. The empirical result of work done by [18] shows the viability of the proposed model. However, construction of ontology in multiple language manually is very expensive in terms of times and

personals. Unlike this, our proposed approach uses an ontology which is expressed only in a single language and to achieve the multilinguality environment a translation approach is used. Finally, the work done by [19] achieved the promising result for classifying documents in significant set of languages, but the classification is not satisfactory as expected; one of the reasons is due to unsatisfactory additional components such as feature extraction, which can be solved together with text classification. Applying dimensionality reduction techniques (i.e. feature selection or feature extraction) is beneficial for the increasing scalability, reliability, efficiency and accuracy of text classification [17].

A main problem to solve for a good classification of documents (texts) is way of representing documents in order to facilitate their processing, and keep only useful information for classification. The most widely used representation in this area is the bag of words representation, where each term or term stem is an independent feature. Works like [19] used a bag of words technique to extract words and represent a given document with all extracted terms. This is not a good approach as explained earlier, however in our investigation a given document is represented with occurrence of lemma words in a document and we call this lemma frequency feature selection method. Since, morphological variations should be considered as equivalent and therefore we integrate a Morphological Analyzer during index term selection, in which different forms of the same word are consolidated into a single word. Hence, in our proposed work, the terms of a document are represented by stem words rather than by the original words and this will enhance our index term selection.

Another issue that makes works on [19] not satisfactory is because in their work for matching of document representative terms with domain ontology concepts used an exact string matching and this minimizes the chance of finding a match between both parties. So, in our investigation in order to alleviate such problem we use different enhanced matching techniques together. First one is, using an exact string matching like the work done on [19], but before concept matching with this technique we perform string preprocessing strategies such as stopword removal, stemming for both sides (i.e. document terms as well as ontology concepts) in order to enhance the probability of finding an exact string match or to decrease number of terms out of the domain ontology. Moreover, the semantical concept matching used a third party knowledge source (i.e. Wordnet) to perform matching based on semantically relation between concepts of both sides (i.e. document representative term along with ontology concept).

Chapter Four

Design of Multilingual Classifier for Under-resourced Language Documents

4.1 Introduction

This chapter describes and discusses the stages of developing of a classifier which can categorize documents written in under-resourced languages on top of ontology built in resourced language. Hence, in this chapter, the development process of this classifier will be described in details.

4.2 Tools used

Before going into the detail of proposed classifier architecture and functioning, we provide some background on the tools that we used to achieve the classification of documents written in under-resourced languages (i.e. Amharic, Afaan Oromo and Tigrinya) using the ontology which is built in resourced language (i.e. English). Hence, in this subsection, a detail of the tools used in the development of proposed classifier will be described.

4.2.1 TreeTagger

TreeTagger 5 [42] is a probabilistic language independent part-of-speech (POS) tagger. It was developed by Helmut Schmid in the TC project at the Institute for Computational Linguistics of the University of Stuttgart. This tool allows annotation of multilingual texts with POS and lemma information and it has been successfully used to tag texts in 17 different languages.

Its major strength is that it is adaptable to other languages if a lexicon and a manually tagged training corpus are available. On the other hand, the not standard definition of this training corpus results into a difficult POS output management. For example, Table 4.1 shows a sample output using the English parameter file.

We can see that, to each word of the original text, is associated the corresponding POS and lemma. The lemma is the canonical form or dictionary form of a set of words and the keyword to identify the POS label (e.g., DT for determiner, JJ for adjective), is not standard but depends on personal choice of the parameter file owner. Hence, in our proposed approach we adopt TreeTagger for Java 6, a Java wrapper around the popular TreeTagger package by Helmut

Schmid; it was written in Java 5 with a focus on platform-independence and easy integration into applications.

Word	POS (Part of Speech)	Lemma
The	DT	The
TreeTagger	NP	TreeTagger
is	VBZ	Be
easy	JJ	Easy
to	TO	To
use	VB	Use

Table 4.1: Sample of treetagger output - English parameter file

As explained earlier, TreeTagger is a tool for annotating text with part-of-speech and lemma information. In this investigation, the POS information of each words of a given text is used for two main functionalities: to disambiguate the words sense generation based on POS information during word by word text translation and to select the lemma of a word during honrmorpho lemma selection functionality. Since, due to lack of lemma information of supported language, we use TreeTagger for POS annotation only.

4.2.2 HornMorpho

HornMorpho is a Python program that analyzes Amharic, Oromo, and Tigrinya words into their constituent morphemes (meaningful parts) and generates words, given a root or stem and a representation of the word’s grammatical structure [82]. It is part of the L3 project at Indiana University, which is dedicated to developing computational tools for under-resourced languages.

This tool relies on a long history of research on Amharic, Afaan Oromo, and Tigrignya gramma, while introducing a few new notions. Moreover, a hornmorpho does not provide a semantic representation of the input word, rather a representation of the grammatical structure of the word. In any case, the grammatical structure would be needed by any system that performs a semantic analysis.

In order to perform morphological analysis, hornmorpho has two functionalities, one for analyzing single word and the other for analyzing all of the words in a file. These functions are anal word and anal file respectively and which takes input words and output a root or stem and a grammatical analysis. In both hornmorpho functionalities before analyzing the input (i.e. single word or file content) the language of a given input should be indicated. In this tool a language mark “am” always indicates for Amharic, “om” for Afaan Oromo and “ti” for Tigrinya.

For example: - to analyze the morphological structure of an Amharic word ‘የማያስፈልጋትስ’ as follows

l3.anal ('am', 'የማያስፈልጋትስ')

The HornMorpho produces the following morphological structure of a word

word: የማያስፈልጋትስ

POS: verb, root: <fl_g>, citation: አስፈላጊ

subject: 3, sing, masc

object: 3, sing, fem

grammar: imperfective, causative, relative, definite, negative

conjunctive suffix: s

As explained earlier, due to lack of lemma information supported languages the TreeTagger module is used only for POS annotation. Hence, lemma information is required for word by word text translation, since bilingual dictionaries are built on top lemma of words. Due to this, in order to get the lemma information of a word hornmorpho is adopted in this investigation.

However, as we have seen in the above example, from the analyzed hornmorpho output a line start with “word:” and “POS:” information only is required to capture the text words as well as root word information.

4.2.3 Jena

Jena is an open source Semantic Web framework for java developed by Brian McBride of HP. It provides extensive Java libraries for helping developers develop code that handles RDF, RDFS, RDFa, OWL and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL and RDFS ontologies, and a variety of storage strategies to store RDF triples in memory or on disk.

The two main packages of the Jena are:

- `com.hp.hpl.jena.rdf.model`: package for creating and manipulating RDF graphs
 - ✓ `Model`: RDF Model.
 - ✓ `ModelMaker`: `ModelMaker` contains a collection of named models, methods for creating new models (both named and anonymous) and opening previously-named models, removing models, and accessing a single "default" Model for this Maker.
 - ✓ `Literal`: RDF Literal.
 - ✓ `Statement`: RDF Statement.
- `com.hp.hpl.jena.ontology`: package that provides a set of abstractions and convenience classes for accessing and manipulating ontologies represented in RDF
 - ✓ `OntModel`: an enhanced view of a Jena model that is known to contain ontology data, under a given ontology vocabulary (such as OWL).
 - ✓ `OntClass`: interface that represents an ontology node characterizing a class description.
 - `ComplementClass`
 - `IntersectionClass`
 - `UnionClass`
 - ✓ `OntResource`: provides a common super-type for all of the abstractions in this ontology representation package.
 - ✓ `OntModelSpec`: encapsulates a description of the components of an ontology model, including the storage scheme, reasoner and language profile.
 - ✓ `ObjectProperty`: interface encapsulating properties whose range values are restricted to individuals (as distinct from datatype valued properties).

- ✓ `DatatypeProperties`: interface that encapsulates the class of properties whose range values are datatype values (as distinct from `ObjectProperty` whose values are individuals).

In this investigation, to manage the ontology we use Jena tool, particularly for two main functionalities: to create and store the ontology model on a memory and also in order to iterate over the ontology concepts based on formulated SPARQL query.

4.2.4 Snowball Porter Stemmer

In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their stem, base or root form, generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. One of widely popular adopted and extended stemmer algorithm is Porter stemmer algorithm.

Porter stemmer is one of the most commonly used truncation stemmers. It removes affixes from a word over a number of iterations until all the rules/conditions are considered. Porter's algorithm was developed for the stemming of English-language texts but the increasing importance of information retrieval in the 1990s led to a proliferation of interest in the development of conflation techniques that would enhance the searching of texts written in other languages. By this time, the Porter algorithm had become the standard for stemming English, and it hence provided a natural model for the processing of other languages. In some of these new algorithms the only relationship to the original is the use of a very restricted suffix dictionary [109], but Porter himself has developed a whole series of stemmers that draw on his original algorithm and that cover Romance (French, Italian, Portuguese and Spanish), Germanic (Dutch and German) and Scandinavian languages (Danish, Norwegian and Swedish), as well as Finnish and Russian [110].

These stemmers are described in a high-level computer programming language, called Snowball [111] that has been developed to provide a concise but unambiguous description of the rules for a stemmer. Some non-English stemmers can operate effectively using simple sets of rules, with

Latin being perhaps the best example of a language that is defined in what is essentially algorithmic form [112]. However, this level of regularity and simplicity is by no means common; in such cases, Snowball provides a concise but powerful description that can then be processed by a compiler to give a C or Java implementation of the algorithm for the chosen language [109]. In general, the analysis of snowball Porter stemmer has shown that its performance is one of the best in terms of IR recall and precision; hence in this investigation a Java version of snowball stemmer version 3.0.3 is adopted. In this investigation, Porter stemmer is used to enhance the matching of ontology concepts along with translated sense by reducing both party words in to their stem form.

4.2.5 Wordnet

Wordnet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms or synsets, each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningful related words and concepts can be navigated. WordNet is also freely and publicly available and its structure makes it a useful tool for computational linguistics and natural processing [106].

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the grouping of words in thesaurus does not follow any explicit pattern other than meaning similarity.

The main relation among words in WordNet is synonymy, as between the words shut and close or car and automobile [107]. Synonyms are words that denote the same concept and are interchangeable in many contexts are grouped into unordered sets or synsets. Each of WordNet's 117,000 synsets is linked to other synsets by means of a small number of "conceptual relations". Additionally, a synset contains a brief definition and in most cases, one or more short sentences illustrating the use of the synset members. Word forms with several distinct meanings are represented in as many distinct synsets. Thus, each form meaning pair in WordNet is unique.

The lexical database WordNet is particularly well suited for similarity measures, since it organizes nouns and verbs into hierarchies of is-a relations. In version 2.0, there are nine noun hierarchies that include 80,000 concepts, and 554 verb hierarchies that are made up of 13,500 concepts.

Is-a relations in WordNet do not cross part of speech boundaries, so WordNet-based similarity measures are limited to making judgments between noun pairs (e.g., cat and dog) and verb pairs (e.g., run and walk). While WordNet includes adjectives and adverbs, these are not organized into is-a hierarchies so similarity measures cannot be applied. However, concepts can be related in many ways beyond being similar to each other. For example, a wheel is a part of a car, night is the opposite of day, snow is made up of water, a knife is used to cut bread, and so forth. As such WordNet provides additional (non-hierarchical) relations such as has-part, is-made-of, is-an-attribute-of, etc. In addition, each concept (or word sense) is described by a short written definition or gloss.

Measures of relatedness are based on these additional sources of information, and as such can be applied to a wider range of concept pairs. For example, they can cross part of speech boundaries and assess the degree to which the verb murder and the noun gun are related. They can even measure the relatedness of concepts that do not reside in any is-a hierarchy, such as the adjectives violent and harmful.

Several methods for calculating semantic similarity between words in WordNets exist and can be classified into three categories:

(I). Edge based Methods

To measure the semantic similarity between two words is to measure the distance or path linking of the words and the position of the word in the taxonomy. Three similarity measures are based on path lengths or distance between concepts: lch [113], wup [114], and path. The lch measure finds the shortest path between two concepts, and scales that value by the maximum path length in is-a hierarchy in which they occur. Wup [114] finds the path length to the root node from the least common subsumer (LCS) of the two concepts, which is the most specific concept they share as an ancestor. This value is scaled by the sum of the path lengths from the individual

concepts to the root. The measure path is equal to the inverse of the shortest path length between two concepts.

(II). Information based Statistics Methods

To solve the difficult problem to find a uniform link distance in edge based methods, Rasnik proposes an information based statistic method [119]. The basic idea is that the more information two concepts have in common, the more similar they are and this approach is independent of the corpus.

There are three popular similarity measures are based on information content: res [115], lin [116], and jcn [117]. The lin and jcn measures augment the information content of the LCS of two concepts with the sum of the information content of the individual concepts. The lin measure scales the information content of the LCS by this sum, while jcn subtracts the information content of the LCS from this sum (and then takes the inverse to convert it from a distance to a similarity measure)

(III). Hybrid Methods

The combination of the above semantic similarity measurement methods (i.e. Edge based methods and Information based statistics methods).

In this investigation, a pure Java API that implements a variety of semantic similarity and relatedness measures based on information found in the lexical database WordNet called WS4J is used [108]. In particular, it supports the measures of resnik, Lin, Jiang-Conrath, Leacock-Chodorow, Hirst-St.Onge, Wu-Palmer, Banerjee-Pedersen, Patwardhan-Pedersen and so on.

Moreover, in this investigation, a WS4J version 1.0.1 is adopted in order to measure the semantic relatedness between the ontology concept and the translated sense. Particularly, a hybrid methods, which is a combination of shortest Path based Measure, Wu & Palmer's Measure, and Lin's Measure semantic related ness measurements are used (for more clarification see chapter 6 under sub section of semantic concept mapping).

4.3 Architecture of Multilingual Documents Classifier

As mentioned in chapter 1, the aim of this research work is to develop a text classifier which can classify under-resourced language documents with resourced language ontology without a language barrier. So, in this section the details of the structural design and features of the proposed text classifier will be presented.

Figure 4.1 shows the general architecture of the implemented framework for Classifier of under-resourced language documents based on resourced language ontology. It is structured into four main modules (i.e. pre-processing module, translation module, mapping module and classification module) based on the data and process flow between the components.

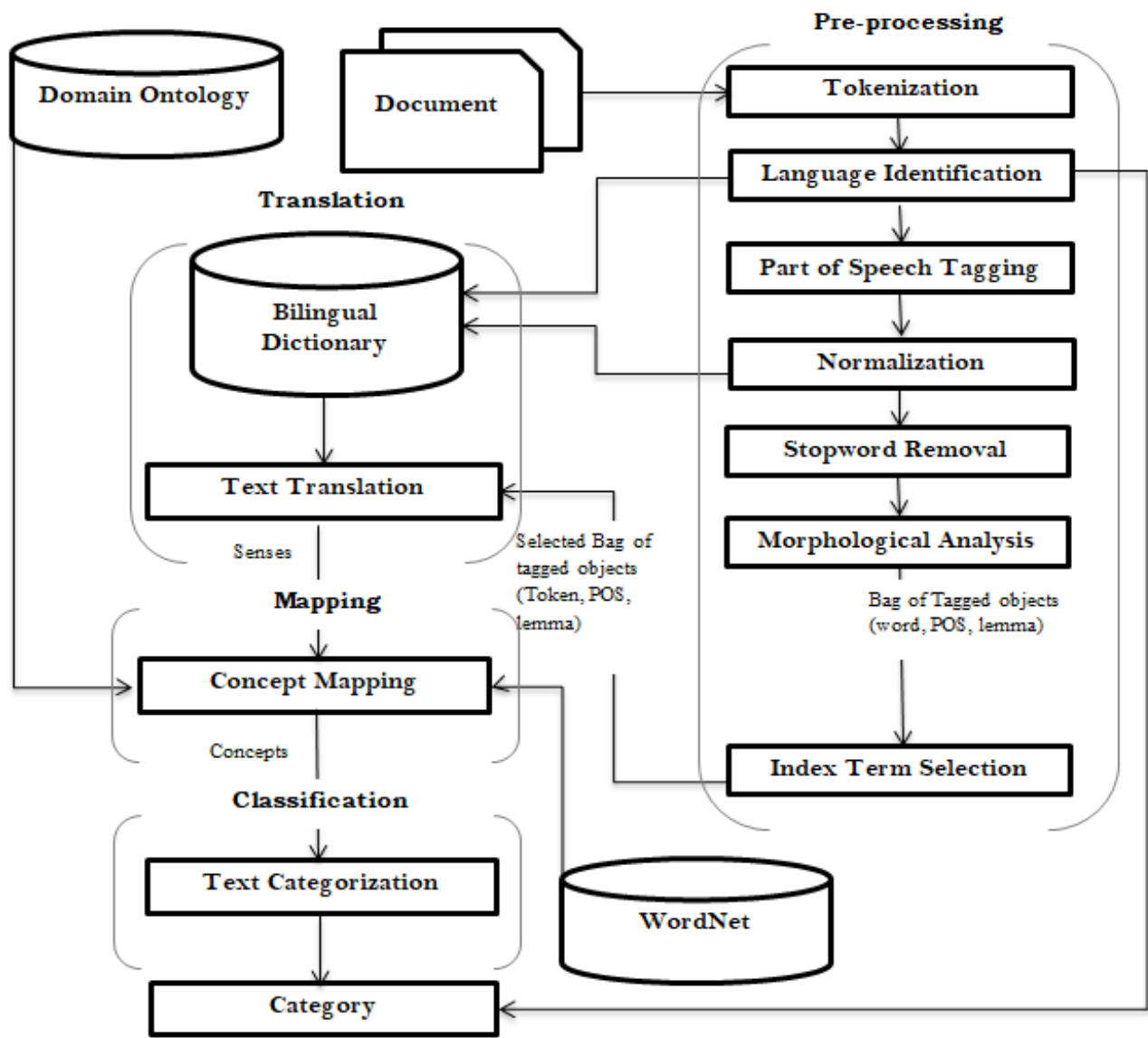


Figure 4.1: Design of Multilingual Classifier for Under-resourced Language Documents

As shown in Figure 4.1, the input to the pre-processing module is a document that is going to be classified by the system. After acting on the input document, the pre-processing module generates a bag of representative tagged objects (each object with word itself, POS and lemma information). Then the pre-processing module requests the translation module to get the corresponding sense in a target language. The translation module checks the bag of representative tagged objects information along the bilingual dictionary and returns the corresponding bag of sense. After that, concept mapping module accepts bag of senses and perform mapping of these translated senses along the ontology concepts.

During each mapping of translated sense along the ontology concept, the concept weighting is computed based on the lemma occurrence in given document. Finally, the classification module accepts bag of weighted concepts and depending on their importance or weight assigned as an actual category of a given document. The final assigned category of a document is represented in actual given document language as well as English language.

In this section we are going to analyze and discuss every module and its sub components of proposed classifier architecture in detail by providing an information explanation and functional description.

4.4 Pre-processing

It is the first module of proposed classifier and it is responsible to accept the input document and produce a set of selected tagged objects (each tokens with POS and lemma information) after carrying out tokenization, language identification, POS tagging, normalization, stopword removal, morphological analyzer and index term selection. In the coming sub-sections, the components of this module are described to show how each components of the module is designed and implemented.

4.4.1 Tokenization

This module is responsible to extract bag of words obtained from the string representation of the input document.

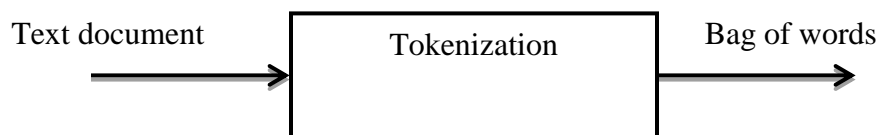
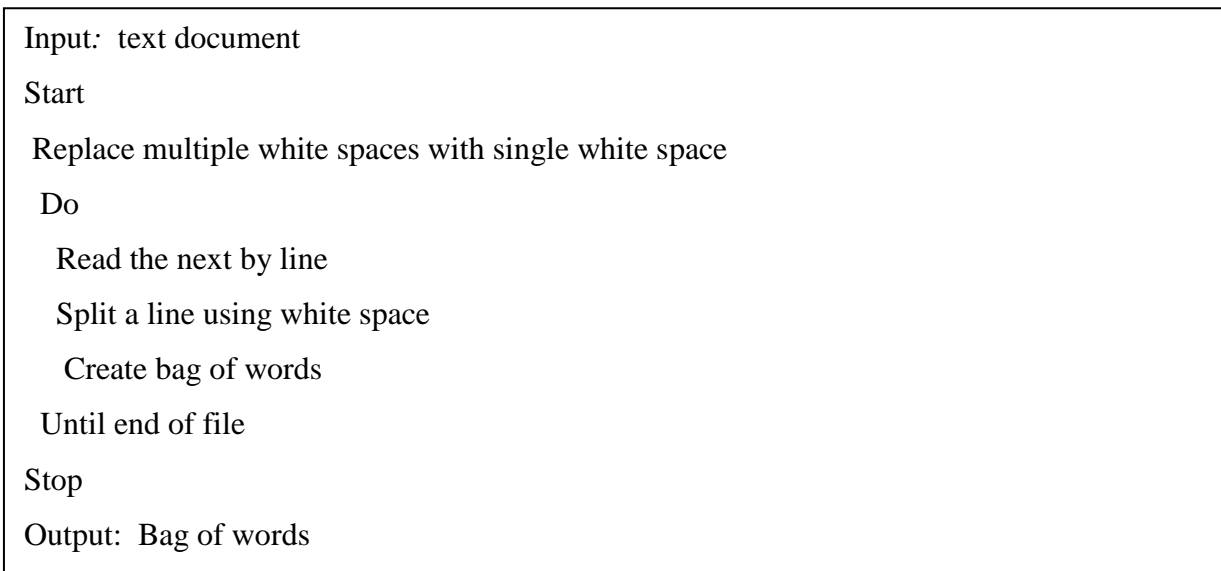


Figure 4.2: Tokenization

In order to extracting bag of words from the input document, all occurrences of multiple white spaces are replaced by a single white space and the input string is split using the white space character.

Algorithm 4.1 shown below is designed to implement the tokenization component to perform splitting of input text document into bag of words. In order to achieve this, the Algorithm 4.1 first reads the input text document line by line until end of file and each line is tokenized into words with whitespace. Finally, all tokenized words are added to bag of words.



Algorithm 4.1: Tokenization

As shown in figure 4.2 the unclassified document text is passed as input to this tokenization component and then bag of words are produced, which is provided to next component of preprocessing module.

4.4.2 Language Identification

This component of preprocessing module is designed in order to recognize the language of the input text document. This component is back bone of the proposed approach, since the modules coming after in the pipeline need to know the language of the input text document to perform the multilingual classification.

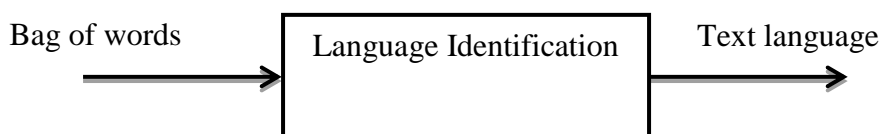


Figure 4.3: Language Identification

There are different types of approaches for language identification of textual document including the character Ngram, words with dictionaries of various languages. However these approaches are computationally unfeasible and an approach that ensures a fast detection of language is based on the use of language stop words as blacklist for language identification [78]. Moreover, the Amharic, Afaan Oromo and Tigrinya languages have their own unique stop words and also the input text for classification is at a document level, hence it can easily identify the language with stopword based approach. Algorithm 4.2 shown below is designed to implement the language identification component of this investigation. The algorithm explains , in order to identify the language of input text document from previous tokenization module the bag of words are feed as input and all words in bag of words are checked along each language stop words. When a word from bag of word found in language list of stopwords, then we increment the counter by one. Then, finally, the language of list of stopwords is assigned as a language of the document if and only it has a definitive advantage over others language list of stopwords.

```

Input: list of stopwords, Bag of words
Start
Read word from bag of stopwords
Index bag of stopwords by word length
For word in Bag of words do
  If word length equal with bag of stopwords index then
    If word in Bag of stopwords then
      Increment Bag of stopword language counter
    End if
  End if
End for
If Bag of stopword language has a definitive advantage over others then
  Return Bag of stopword language as text document language
Else
  Return error Message "Unsupported Language"
Stop
Output: Text language

```

Algorithm 4.2: Language identification

In order to speed up the matching process of the words in a bag of words against list of stopwords of each supported language, we used a word length per each language to index the list of stopwords. This minimizes the time taken to search a given word in a list of stopwords for each language. If the language is identified for a given text document, then it proceeds to next component of preprocessing module, otherwise an error message (Unsupported Language) is printed and the operation is not proceeding further.

4.4.3 Part of Speech Tagging

Once the tokenization is done and language of text document is identified, we proceed to part of speech tagging. The tagging component involves in determining the part of speech tag for each words in bag of words provided by tokenization component.

In this phase we adopted TreeTagger, the words are represented by word itself, lemma and part of speech tag (POS). However, in this study due to lack of lemma information, we adopted the hornmorpho in order to generate the lemma of each words. In detail we will discussed on the next section about the adopted morphological analyzer.

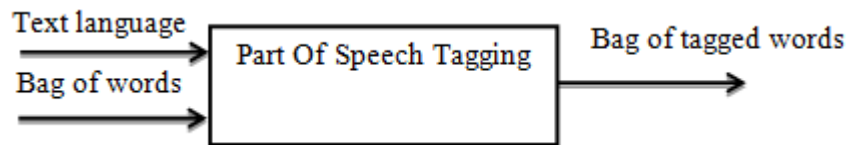


Figure 4.4: Part of speech tagging

As explained in this component, we adopted TreeTagger for Java (tt4j) that is a Java wrapper around the popular TreeTagger package [42]. We trained this tagger by using a manually tagged training corpus for each language (i.e. Amharic, Afaan Oromo and Tigrinya), the detailed explanation of training corpus used in to adopt TreeTagger in this investigation will be explained in chapter 5.

In order to load the correct TreeTagger training file parameter, we use the information about the language identified for text document in the previous step, in particular its name; for that reason, the TreeTagger model must be stored with a format of: [name of language]-utf.par. The outcome of this tagger component contains each word with their POS information. The POS tag of the word is used to disambiguate the lemma selection for Morphological Analyzer and also to disambiguate the meaning of the word during word by word translation. For example the lemma

can appear more than once in our bilingual dictionary having different meaning and in order to disambiguate such semantic ambiguity, a POS is used as a core feature in this study. In detail we will discuss about use of this POS feature under the section of Morphological analyzer and text translation components.

In order to adopt a TreeTagger, we used a training corpus having different POS representation for each particular language and this would be different POS outcome for each TreeTagger corpora. When a POS outcome is different depending on the text language, it is very complex for our text translation operation.

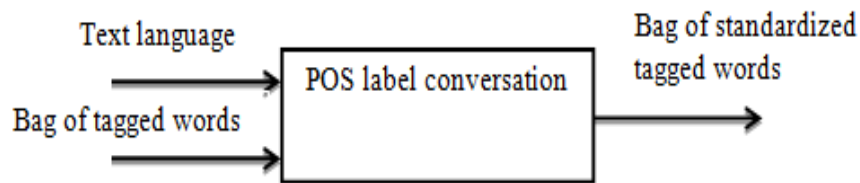


Figure 4.5: POS label conversion

So, in order to handle this complexity, we include a module which can convert the original tagged POS labels of each word into the standard POS labels of the system like listed in Table 4.1, which are required by our bilingual dictionary. Algorithm 4.3 illustrates how to convert the original POS label of each word after POS tagging operation into standardized POS label. Once the text language is identified and the tagging operation is done, the configuration file which contains the original POS label with corresponding standardized POS label is loaded particularly for identified text language.

```

Input: Text language, Bag of tagged words
Start
Read POS configurations file for identified text document language
  For POS label in Bag of tagged words do
    If POS label is in POS configuration file then
      Convert the POS label by corresponding standard POS label
    End if
  End for
Stop
Output: Bag of standardized tagged words
  
```

Algorithm 4.3: POS label conversion

To achieve such POS label conversation, tagged words in bag of tagged words are feed as input to POS label conversation module and the original POS label a word is checked within POS configuration and replace with standardized POS label.

Table 4.2 below shows a standardized POS configuration for Amharic textual language, in order to build a better translation module for our multilingual text classifier.

Orgional POS label	Standardized POS label	POS
vn	n	Noun
np	n	Noun
nc	n	Noun
npc	n	Noun
pron	n	Noun
pronp	n	Noun
pronc	n	Noun
pronpc	n	Noun
aux	v	Verb
vrel	v	Verb
vp	v	Verb
vc	v	Verb
vpc	v	Verb
adj	a	Adjective
adjp	a	Adjective
adjc	a	Adjective
adjpc	a	Adjective
adv	av	Adverb

Table 4.2: POS label configuration for Amharic text language

4.4.4 Normalization

As showed in figure 4.1 above, this component is dedicated once the POS tagger component was processed. This component mainly performs two normalization operations. The first responsibility of this module, the normalization of homophones that follows, language writing system that has homophone characters.

For example, in Amharic language it is common that the character ስ and ሥ are used interchangeably as ስሬ and ሥሬ to mean “work”. Such type of inconsistency in writing words is handling by replacing characters of the same sound by a common symbol. These characters cause unnecessary increase in the number of document representative words that causes large data size processing.

In order to handle this issue, we have created the configuration file for each language where the homophone characters are converted to common symbols. This module tries to reduce the number of document representative words that causes large data size processing.

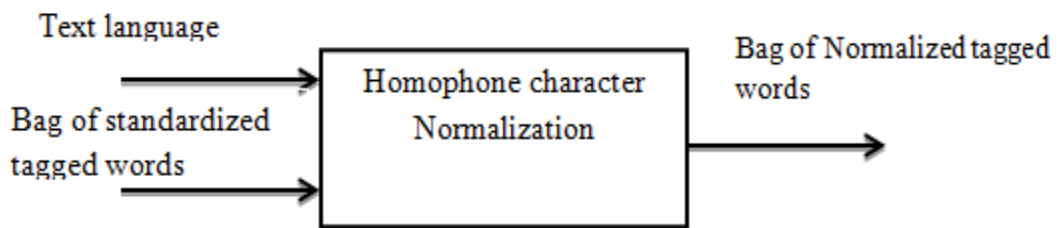


Figure 4.6: Homophone character normalization

For example, the different forms of an Amharic word ‘Hailu’, which are ሀይሉ, ሃይሉ, ሐይሉ, and ኃይሉ are all converted to the common form ሀይሉ by changing the first character of the three words. Table 4.3 shows sample character replacements used in [81]. In this investigation, normalization is used for both Amharic and Tigrinya textual documents with the same list of normalized character value.

Homophone Characters	Normalized Characters Value
ሐ፣ ሃ፣ ኃ፣ ሄ፣ ሐ	ሀ
ዐ፣ ዓ	አ
ሠ ሠ፡ ፡ ፡ ሠ	ሰ ሰ፡ ፡ ፡ ሰ
ከ፡	ኮ
ኀ	ኀ፡
ወ፡	ወ፡

Table 4.3: Sample of Amharic and Tigrinya Normalized

Algorithm 4.5 designed to implement the homophones character normalizer component of this investigation. The text language and bag of expanded tagged words is feed as input and each words character from this bag is checked along the homophems character configuration file and if found the homophems character is replaced with corresponding normalized character.

```

Input: Text language, Bag of expanded tagged words
Start
Read homophones character configurations file for identified text document language
For word in Bag of expanded tagged words do
  If word is in homophones character configurations file then
    Replace homophones character of a word with corresponding normalized character
  End if
End for
Stop
Output: Bag of normalized tagged words

```

Algorithm 4.4: Homophone Characters Normalization

The next operation of this component is word expansion. The word expander accepts a word as a sequence of characters. The word expander first checks whether a word is abbreviated or not, if a word contains forward slash (“/”) or period (.) then it is considered as abbreviated form and checked along the list of abbreviated word of identified text language to return the corresponding expanded form of the word. The final result of this module is an expanded form of a word. For example in Amharic language, ትምህርት ቤት can be written as ት/ቤት, ዶክተር as ዶ/ር and similar for other languages also.

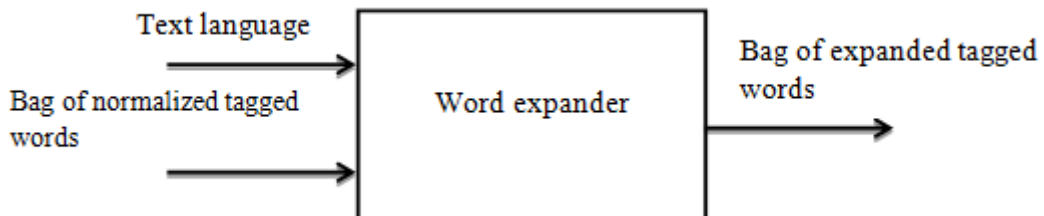


Figure 4.7: Word Expander

Algorithm 4.4 is designed to implement the word expander component. This algorithm takes

```
Input: Text language, Bag of normalized tagged words
Start
Read abbreviated word configurations file for identified text document language
  For word in Bag of standardized tagged words do
    If word contains"/" or "." then
      If a word in Bag of abbreviated word then
        Replace abbreviated word with corresponding expanded word form
      End if
    End if
  End for
End if
End for
Stop
Output: Bag of expanded tagged words
```

Algorithm 4.5: Word Expander

the identified text document language and Bag of standardized tagged words as input and identify an abbreviated word by checking if a word contains “/” or “.” or not. If a word contains one of these characters, then it is looked up along the list of abbreviated words to replace with the corresponding expanded word form.

In order to execute the above two normalized operations, the identified text language and bag of normalized tagged word should be passed as a parameter.

Beside this, in order to find a match during lemma generation of a word along the HornMorpho analyzed output (for more clarification see under this chapter of Morphological Analyzer component), the homophone character normalized value used is similar with HornMorpho character normalized value.

4.4.5 Stopword Removal

Words which are deemed irrelevant or non-content bearing words for text classification purpose should be removed and these words are called stopwords. These words occur most frequently in documents, but have no relevance or no impact to discriminate document category. For Instance

in Amharic words like ነው, ቢሆን, ስለ are non-content bearing words or stopwords. List of stop words of all supported languages (i.e. Amharic, Afaan Oromo and Tigrinya) are listed in Appendix A.

Such frequently occurring words are generally used to “glue” sentences together but they usually do not carry meanings. In this study this removal process is normally done automatically by comparing list of words generated for input text document with words in a stopwords list. In order to enhance the efficiency of matching words in both collections we used length of a word as index such as words with ‘Amharic’ text language and having length of four only compares with Amharic stopword list having four word lengths, this speed up the stopword detection process.

From our study, applying stopword removal enhances efficiency of our Morphological Analyzer, since the number of words which are analyzed by this component is only those relevant words for our classification purpose. In addition to this, it also reduces the complexity of the document representation and reduces the number of words to be processed for the upcoming modules of classifier.

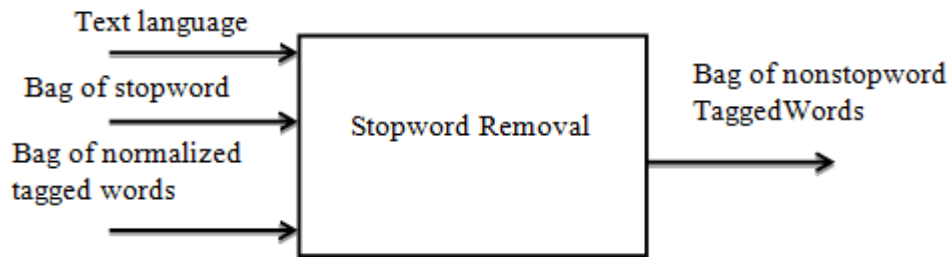


Figure 4.8: Stopword Removal

As shown in Figure 4.8, in order to remove stopwords, the language of text document should be identified first to load stopword list file for that particular text language. All bag of tagged words which were normalized from previous component feed to this component and all stopwords are filtered and removed by comparing the normalized tagged words against the loaded stopwords word list.

As shown in Figure 4.1, in order to enhance the detection of stopwords for a word in bag of words, the normalization operation is done along the list of stopwords and a word in bag of words before the stopword removal is done. Since, homophone characters in both parties are represented by a common character symbols.

As clarified in Algorithm 4.6, the text language and bag of normalized tagged words given as input and a bag of stopwords for identified text language which is indexed with word length is

loaded. In order to check all words in bag of tagged words are stopword or not, first the word length is computed and a word is checked along a bag of words having equal index word length. If a word is not found in bag of stopword then it is stored in bag of non stopword tagged words unless it is ignored.

```
Input: Text language, Bag of normalized tagged words
Start
Read list of stopwords for the identified language
For each word in Bag of normalized tagged words do
  If word length equal with bag of stopwords index then
    If word is in bag of stopwords then
      Continue
    Else
      Add to bag of non stopwords
  End if
End if
End for
Stop
Output: Bag of nonstopword tagged words
```

Algorithm 4.6: Stopword Removal

4.4.6 Morphological Analysis

The lemma identification is fundamental component in order to reach our multilingual document categorization goal. Most of the word contents of a bilingual dictionary are constructed in lemma form, hence the bilingual dictionary is usually able to find and translate words correctly in lemma form. To achieve this, we adopt a hornmorpho version 2.5 [82]. However, a HornMorpho is a python based tool and our framework is implemented in Java, so in order to integrate this tool to our classifier we use a convenience built in method called `java.lang.Runtime.exec` library in order to execute the python script as parameter [83].

To minimize processing time of hornmorpho, we analyze all tagged words which are provided from previous phase, instead of analyzing a single word at a time. Since, to analyze a single word at a time always HornMorpho needs to load its model for a target language and this is costly and inefficient. In order to enhance such efficiency issue we use a file with read and write operation as a mediator. All lists of words are written along the file from our prototype configuration directory and is passed as a parameter to `l3.anal_file ('textLanguage','inputfile','outputfile')` , is a module of HornMorpho which analyze all the words in an input file and write the analysis to another file.

```
Input: HornMorpho Analyzed output file
Start
Count = 0
Do
  Read the content of HornMorpho analyzed output file line by line
  If line start with "word:" or "?word:" then
    Count +=1
    Save count and line information
  Else
    Save count and line information
  End if
While end of file
Stop
Output: Bag of indexed HornMorpho output
```

Algorithm 4.7: HornMorpho analyzed output

Hence, for test document we create an input file which contains all lists of pre-processed bag of words and also generate HornMorpho analysis to another file as output file. In order to suitable to be accessed later, the HornMorpho analyzed output should be indexed on a memory. As explained earlier, in this investigation the actual text word, POS and lemma information is extracted from hornmorpho analyzed output.

Input: Bag of indexed HornMorpho output

Start

For each indexed word information in Bag of indexed HornMorpho output do

For each information in indexed word information do

If information size > 1

If information starts with “word:” then

Split information by “:”

Token = information [1]

Else if information starts with “?POS:” or starts with “POS:” then

Split information by “,”

Split information [0] by “:” to extract POS

If information contains “stem:” then

Split information [1] by “:” to extract stem

Else if information contains “citation:” then

Split information [2] by “:” to extract citation

End if

End if

Else

If information starts with “?word:” or starts with “word:” then

Split information by “:” and assign information [1] as stem

Assign NULL to POS

End if

End for

End for

Stop

Output: Bag of indexed lemma Information

Algorithm 4.8: Lemma Information per word indexing

Once algorithm 4.7 done, indexing of HornMorpho analyzed output as whole per word information and then algorithm 4.8 extracts only word with bag of POS and lemma information

from output of algorithm 4.7. Hence, to prune other unnecessary information for this study we design an algorithm 4.8, which illustrates to index the word with bag of POS and lemma information.

Algorithm 4.8 depends on the analyzed output of algorithm 4.7, which indexes line of HornMorpho analyzed output per word. Since, the HornMorpho analyzed output can be contain a word only or one or more lemma along with corresponding POS information. But, POS information can also not present. Beside, this to separate the words analyzed grammatical information a new line is included.

Once an indexed bag of lemma information for each bag of tagged word is created, we can proceed to use this information for our translation module as well as index term selection module.

```
Input: Text language, Bag of nonstopword tagged words
Start
Read HornMorpho analyzed output
For tagged word in Bag of nonstopword tagged words do
  If word and POS label is match in HornMorpho analyzed output then
    Return corresponding lemma
  Else if word is match in HornMorpho analyzed output then
    Return all corresponding lemma
  Else
    Return a word itself as lemma
End if
End if
End for
Stop
Output: Bag of tagged objects
```

Algorithm 4.9: Lemma extraction for translation

We designed algorithm 4.9, which illustrates to extract lemma information of a given word to enhance our word by word text translation

(for more explanation see coming sub section of translation module). In general, once the previous component of preprocessing module is done,

```
Input: Text language, Bag of nonstopword tagged words
Start
Read HornMorpho analyzed output
For tagged word in Bag of nonstopword tagged words do
  If word and POS label is match in HornMorpho analyzed output then
    If return more than one lemma then
      For lemma in bag of lemma
        Pick and return first best lemma
      Break
    End for
  Else
    Return lemma
  Break
End if
Else if word is match in HornMorpho analyzed output then
  If return more than one lemma then
    For lemma in bag of lemma
      Pick and return first best lemma
    Break
  End for
  Else
    Return lemma
  Break
End if
End if
End if
Stop
Output: Bag of tagged objects
```

Algorithm 4.10: Word lemma selection for index term selection

we have preprocessed bag of tagged words and text language, which are useful for our Morphological analyzer component.

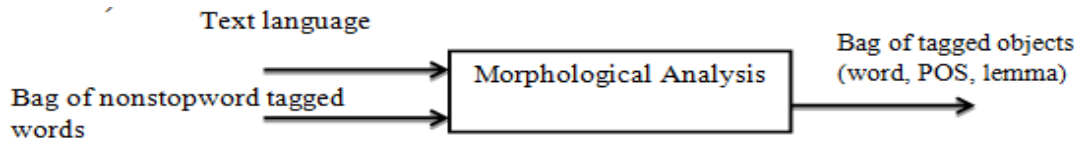


Figure 4.9: HornMorpho combiner

So, the Morphological analysis identifies lemma information of each tagged words in order to achieve lemma based index term selection as well as word by word translation.

The lemma information of a word is useful for the next index term selection module of the proposed approach. Since, the index term selection considers the lemma frequency in a text document. Hence, lemma selection for each tagged word is required and to achieve this in bag of tagged words we have token as well as POS information. On the other hand, along the HornMorpho analysis output we have also token, POS and lemma information.

Therefore, for ambiguous words, we choose the first best lemma analysis as HornMorpho analyses are ordered by their estimated frequency. Algorithm 4.10 illustrates how to select the first best lemma of a given word from analyzed HornMorpho output and assign the selected lemma as value of word lemma information. The selection of lemma is based on different criteria or condition as we have explained in Algorithm 4.10 and finally one of the lemma which fulfils the criteria assigned as lemma of a word.

4.4.7 Index Term Selection

In this study, words that have the capability to represent the given document are selected using term frequency. Term frequency in this work is the number of times lemma of a word occurs within actual text document and we call it lemma frequency. This lemma frequency is used to determine which word was sufficiently significant to represent the document.

Taking an occurrence of an actual word of a document in order to select a word as index term is not a good approach, particularly for a language with big morphological complexity like Amharic, Afaan Oromo and Tigrinya.

Since, usually equivalent words are represented with their inflected word. For example, during term representation of a given Amharic document when “ተማሪ” (student) and “ተማሪዎች”

(students) words are captured as different words, this degrades performance of index term selection as well the text categorization. Hence, in order to eliminate such adverse effects, we are using the lemma information which is assigned by Morphological Analyzer module for word frequency computation. This technique reduces the problem of considering inflected words as different words during term frequency computation. In general, the lemma frequency of a word shows usefulness of a lemma word in the document.

Lemma frequency $LF(d, l)$ is the number of times a lemma occurs in the text document and is defined as

$$LF(d_i, l_k) = \sum_{j=1}^n fl_{jk} \tag{4.1}$$

Where d_i is the i^{th} document, l_k is k^{th} lemma of document d_i and $\sum_{j=1}^n fl_{jk}$ is sum of lemma occurrence l_k in a document d_i .

The Bag of objects which satisfies the threshold value of the lemma frequency based on equation 4.1 is selected as representative object of the given textual document.

In this component, the weight of a tagged object is equal to the lemma occurrence of an object in a text document and this weight is known as lemma frequency and it is associated with tagged object.

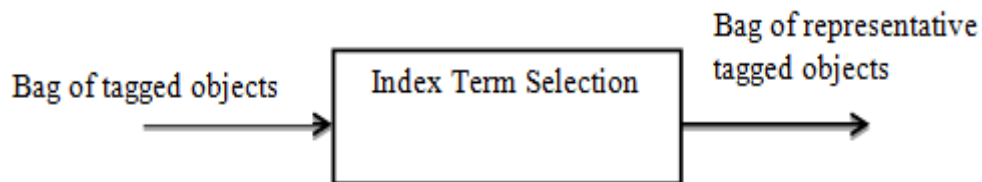


Figure 4.10: Index Term Selection

4.5 Translation

It is the second module of proposed multilingual text classifier and it is responsible to accept the bag of tagged objects information from previous pre-processed module and produces translated senses of the target language (i.e. English). This module has only one component called text translation. In this section we will briefly discuss how the text translation sub component operates and implemented.

4.5.1 Text Translation

The basic idea behind the proposed text classifier operates on a multilingual environment without any language barrier is due to text translation component. Text translation is dedicated to translate representative tagged objects information into English senses. This operation relies on a bilingual dictionary that is in essence mappings between text language tagged objects (token, POS and lemma) with their corresponding English senses.

In this investigation, dictionary is used as knowledge source for our word by word translation task; this dictionary is a bilingual dictionary which contains words in any under-resourced language (i.e. Amharic, Afaan Oromo, and Tigrinya) with their part of speech tag and with the equivalent meaning word in English language. This bilingual dictionary is loaded for a particular text language after a document language is identified with previous language identification component.

As we used a normalization operation during preprocessing module for tagged objects of text document, we also include this preprocessing operation to normalize the bilingual dictionary words when loaded. This normalization operation enhances the matching probability of terms along the bilingual dictionary, since both tagged object information as well as the bilingual dictionary words are normalized with the same operation.

The text translation component needs the bilingual dictionary for identified text language, the actual word with lemma and POS information in order to perform word by word sense generation for a target ontology language i.e. English language.

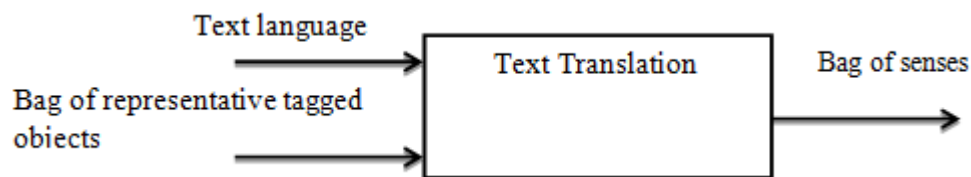


Figure 4.11: Text Translation

To generate the sense from a bilingual dictionary using only lemma and POS information is not enough, because these tagged object information's may not found always. So, to enhance the

probability of sense generating for all tagged objects, the following four alternative text translation functionalities are performed.

- (a) Find and match a word having lemma and POS information
- (b) Find and match an actual word and POS information
- (c) Find and match a lemma without POS information
- (d) Find and match an actual word without POS information

To translate a representative tagged objects information using bilingual dictionary as knowledge source, the above alternative translation functionalities are executed sequentially. Since all tagged objects have word itself, POS and lemma formation, so in this translation module we try to use all these information's independently and in group to enhance our finding of sense from bilingual dictionary. As displayed in Figure 4.12, first a lemma with POS information is used for our text translation purpose and if an equivalent sense for a target language is found so sense is created. Otherwise, an actual word with POS information is used to generate a sense and if found again sense is created.

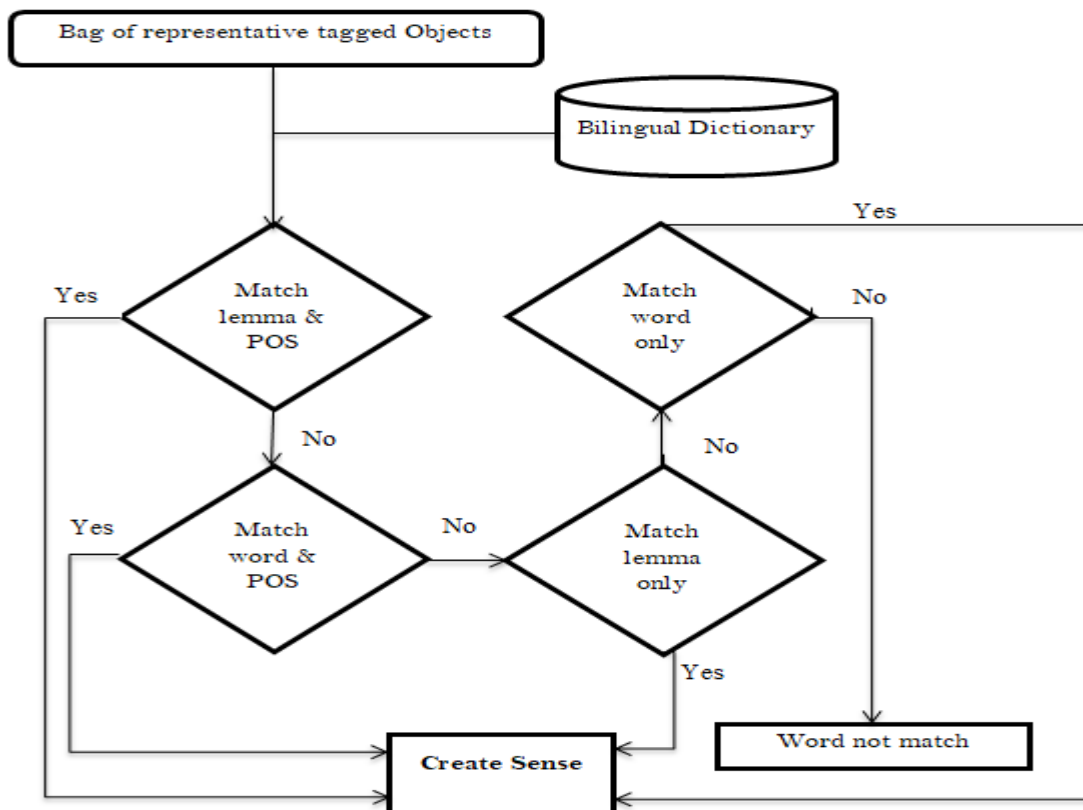


Figure 4.12: Translation of Bag of tagged objects to equivalent sense of target language

But, this information may not find a match in a dictionary. So, a translation uses lemma only as well as word only information respectively and sequentially tries to find a corresponding sense for a target language. As explained earlier, the lemma of a given word may be more than one.. Hence, during text translation each lemma in bag of lemma match along with bilingual dictionary and first most matched word sense is used.

During text translation, clearly a matched word with in a bilingual dictionary can appear in more than once. So, in order to disambiguate such ambiguity during our sense generation, we use POS label features of the word of document. Moreover, all words associated with each matching term in our bilingual dictionary of target language (English language) can be more than ones and each sense are retrieved sequentially until the ontology concept is matched (in detail we will discussed under concept mapping section).

Text translation operation is repeated for each word and it is time consuming process, so in this study to minimize such efficiency issue the following enhancement techniques are employed.

- (a). Removing all the stopword before the translation operation is done. Hence, translation is only done for these non stopwords (i.e. words which are content bearing for text classification task) only.
- (b). Even if words are appear multiple times in the original text document, the translation for these words done once.

4.6. Concept Mapping

Once the text translation generates senses, the next step as shown in Figure 4.1, is mapping of senses to ontology concepts. The concept mapping needs ontology as a knowledge base in order to map the translated sense along the ontology concepts.

In this investigation the domain ontology is built on top of domain concepts extracted from World News domain ontology (WNO) done by [99]. It is based on metadata files created for every single world news HTML web page. The World News Ontology was developed using logic programming as the basic way of data representation and it was implemented using XML.

The metadata representation is based on NewsML (newsml.org). NewsML is an XML based standard developed by International Press Telecommunications Council (IPTC) to represent and manage news throughout its lifecycle including production, interchange and consumer use. NewsML provides set terms for the news domain. This set of terms also known as Newscodes

includes a hierarchy of terms and concepts that can be used to describe news in any field of interest. This hierarchical structure or taxonomy shown in Figure 4.13 consists of three levels:

- **Subject:** topics at this level provide a description of the editorial content of news at a high level
- **Subjectmatter:** a Subjectmatter provides a more precise description
- **Subjectdetail:** provides the most specific description compared to the higher levels

News Ontology Schema

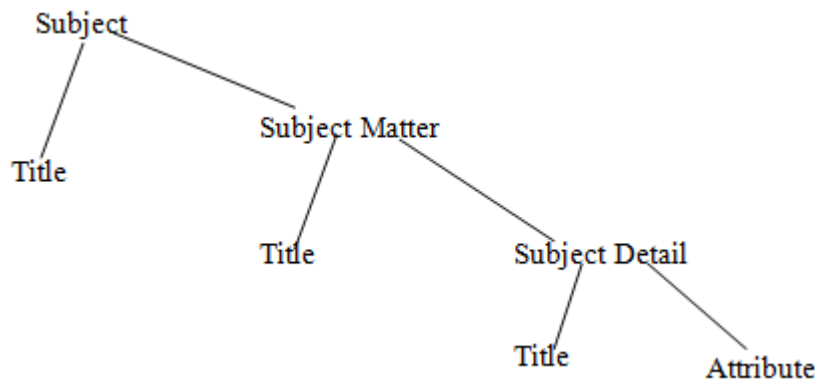


Figure 4.13: Hierarchical structure of Newscodes

The authors of this ontology studied a large number of international news articles from news agency websites and as a result based the ontology on 11 subjects which they felt were sufficiently representative in the domain of world news. Logic Programming (LP) is used to express the ontology and the metadata, which is then transformed into XML format. With LP, new rules can be added allowing processing and reasoning by a LP language.

Figure 4.14 shows a sample extract of the xml definitions for the above three levels of the ontology.

```
<ONTOLOGY>
<SUBJECT>
<TITLE>crime_law_justice</TITLE>
<SUBJECTMATTER>
<TITLE>crime</TITLE>
<SUBJECTDETAIL>
<TITLE>murder</TITLE>
```

```

</SUBJECTDETAIL>
<SUBJECTDETAIL>
<TITLE>computer_crime</TITLE>
</SUBJECTDETAIL>
<SUBJECTDETAIL>
<TITLE>theft</TITLE>
</SUBJECTDETAIL>
•
•
</SUBJECTMATTER>
<SUBJECTMATTER>
<TITLE>judiciary</TITLE>
</SUBJECTMATTER>
<SUBJECTMATTER>
<TITLE>lawyer</TITLE>
<ATTRIBUTE>name</ATTRIBUTE>
<SUBJECTDETAIL>
<TITLE>judge</TITLE>
<ATTRIBUTE>name</ATTRIBUTE>
</SUBJECTDETAIL>
<SUBJECTDETAIL>
<TITLE>court_administration</TITLE>
</SUBJECTDETAIL>
</SUBJECTMATTER>
</SUBJECT></ONTOLOGY>

```

Figure 4.14: Sample extract taken from the WNO

The news category used in this investigation like agriculture, business and economics, crime, education, health, science and technology, sport with sub category of football and athletics extracted from entire WNO ontology was used for our experiments. WNO chosen because it is fairly small in size, which is manageable and it felt unnecessary to prune it. In addition this, it is simplistic because it only shows an IS-A hierarchy between nodes of the ontology. As well, in

WNO class information is given in a top down fashion, and it is very easy to extract domain concepts to build our news domain ontology.

Once the translation module is done and the ontology model is loaded on a memory, the concept mapping is devoted to provide a bag of concepts as shown in Figure 4.15.

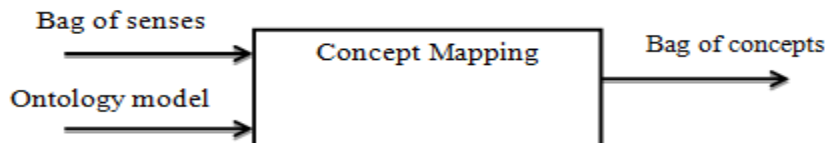


Figure 4.15: Concept Mapping

Mapping concepts of ontology for a given set of translated senses is not trivial. We have to deal with lexical as well as semantic relationship between both parties. This component performs lexical and semantic mapping between the ontology concepts and the senses provided by the previous text translation component.

4.6.1. Lexical Mapping

Generally in ontologies, the class names are usually enclosed in `rdf: label` tags. A label could be represented as a string of characters which could be a term or else a group of words. They are used to afford a human readable explanation of the classes and are distinctive in ontologies.

The lexical or syntactic matching begins the matching process by calculating string matching between the labels of the class of the input ontology with translated senses. In this study an exact string matching is used, which finds an exact string matching between translated sense and ontology concept is used along with string preprocessing strategies such as stopword removal, stemming. Such preprocessing tasks are used to increase the probability of finding an exact match between the translated senses and the ontology concepts. In the first preprocessing task, word constituent matching stop words such as 'a', 'the', 'of', 'in', etc. are dropped from multi-word terms. Remaining words for each term are compared through exact string matching. After performing this phase, words like 'meeting-place' and 'meeting-of-place' can be matched.

In the second preprocessing task, we adopt the Snowball stemmer for English language [86]. This stemmer does not need to be identical to the morphological root of the term; it is usually

sufficient that related words map to the same stem for both ontology concept and sense, even if this stem is not a valid root.

4.6.2. Semantical Mapping

Linguistic features remain essential for developing a primary set of alignments which can be refined by using other types of matching. Even though, the lexical string matching provides most essential clues to check whether the ontology concept and senses are identical or not; it is important to discover semantic relations between them based on various descriptions attached to them. In this study in order to accomplish this we used third party knowledge source i.e. WordNet.

After matching the concepts along senses lexically it is necessary to execute matching based on some background knowledge sources. Since the entities in both parties can be expressed using different terms, the matching between the entities that are semantically related cannot be found. This problem can be solved by matching two parties using WordNet of English language, which aids in finding the semantic similarity between the entities even if those entities are lexically or structurally not overlapped.

As explained earlier, WordNet is a large repository of English items, has been used to provide these semantical relations. This kind of mapping is complementary to the pure string similarity metrics. There are cases where lexical metrics fail to identify similarity between strings that are terminologically different but semantically similar. For example “student” and “learner” are semantically similar although they are lexically distant from each other. Hence, in order to consider such semantic relationship between the ontology concepts and translated sense we adopt the combination of three WordNet semantical similarity measures proposed by [119], which considers WordNet path link as well as information content information.

I. Shortest Path based Measure

The measure only takes $len(c1, c2)$ into consideration. It assumes that the $sim(c1, c2)$ depend on how close the two concepts are in the taxonomy. In fact this measure is a variant on the distance method [86, 87]. It is based on two observations. One is that the behavior of conceptual distance resembles that of a metric. The other is that the conceptual distance between two nodes is proportional to the number of edges separating the two nodes in the hierarchy [91].

$$simpath(c1, c2) = 2 * deep_max - len(c1, c2) \quad (4.2)$$

From formula (4.2) it is noted that,

(a) For a specific version of WordNet, *deep_max* is a fixed value. The similarity between two concepts (*c1*, *c2*) is the function of the shortest path *len* (*c1*, *c2*) from *c1* to *c2*.

(b) If *len(c1,c2)* is 0, *simpath(c1,c2)* gets the maximum value of $2 * \text{deep_max}$. If *len(c1,c2)* is $2 * \text{deep_max}$, *simpath* (*c1,c2*) gets the minimum value of 0. Thus, the values of *simpath* (*c1*, *c2*) are between 0 and $2 * \text{deep_max}$.

(c) *len*(mail, vehicle) = *len*(self-propelled vehicle, bicycle) = 2, therefore, *simpath* (mail,vehicle) = *sim path* (self-propelled vehicle, bicycle).

II. Wu & Palmer's Measure

Wu and Palmer introduced a scaled measure [89]. This similarity measure takes the position of concepts *c1* and *c2* in the taxonomy relatively to the position of the most specific common concept *lso* (*c1*, *c2*) into account. It assumes that the similarity between two concepts is the function of path length and depth in path-based measures.

$$\text{simWP} (c1, c2) = \frac{2 * \text{depth} (\text{lso}(c1, c2))}{\text{len}(c1, c2) + 2 * \text{depth} (\text{lso}((c1, c2)))} \quad (4.3)$$

From formula (4.3) it is noted that,

(a) The similarity between two concepts (*c1*, *c2*) is the function of their distance and the lowest common subsume (*lso* (*c1*, *c2*)).

(b) If the *lso(c1,c2)* is root, $\text{depth}(\text{lso}(c1, c2)) = 1, \text{simWP}(c1, c2) > 0$; if the two concepts have the same sense, the concept *c1*, concept *c2* and *lso(c1,c2)* are the same node. $\text{len} (c1, c2) = 0. \text{simWP} (c1, c2) = 1$; otherwise $0 < \text{depth}(\text{lso}(c1, c2)) < \text{deep_max}, 0 < \text{len}(c1, c2) < 2 * \text{deep_max}, 0 < \text{simWP} (c1, c2) < 1$. Thus, the values of *simWP* (*c1*, *c2*) are in [0, 1].

(c) *len* (mail, bicycle) = *len* (wheeled vehicle, bus) = 4, and *lso*(mail, bicycle) = *lso*(wheeled vehicle, bus) = conveyance, therefore *simWP*(mail, vehicle) = *simWP* (self-propelled vehicle, bicycle).

III. Lin's Measure

It assumed that each concept includes much information in WordNet. Similarity measures are based on the Information content of each concept. The more common information two concepts share, the more similar the concepts are. Lin proposed the following method for similarity measure [90].

$$\text{simlim}(c1, c2) = 2 * \frac{\text{IC}(\text{Iso}(c1, c2))}{\text{IC}(c1) + \text{IC}(c2)} \quad (4.4)$$

It uses both the amount of information needed to state the commonality between the two concepts and the information needed to fully describe these terms.

From formula (4.4) it is noted that,

(a) The measure has taken the information content of compared concepts into account respectively. As $\text{IC}(\text{Iso}(c1, c2)) \leq \text{IC}(c1)$ and $\text{IC}(\text{Iso}(c1, c2)) \leq \text{IC}(c2)$, therefore the values of this measure vary between 1 and 0.

(b) $\text{Iso}(\text{mail}, \text{bicycle}) = \text{Iso}(\text{bicycle}, \text{school bus}) = \text{conveyance}$; if $\text{IC}(\text{mail}) = \text{IC}(\text{bicycle}) = \text{IC}(\text{school bus})$, then $\text{simLin}(\text{mail}, \text{bicycle}) = \text{simLin}(\text{school bus}, \text{bicycle})$.

In this investigation, the chosen methods are selected and used in combination in order to ensure two principals roles: The first one calculates distance between two words and their positions in the taxonomy with the methods Wup and Path. The second role calculates the probability of the word's appearance in the taxonomy based on information theory with Lin's method. The aggregate formula is:

$$\text{sim}(e1, e2) = \frac{\sum \text{simPath} + \text{simWup} + \text{simLin}}{3} \quad (4.5)$$

The overall pictorial description of concept mapping module is described in Figure 4.16. As explained, to map a term with ontology concept, the above ontology mapping methods are executed sequentially.

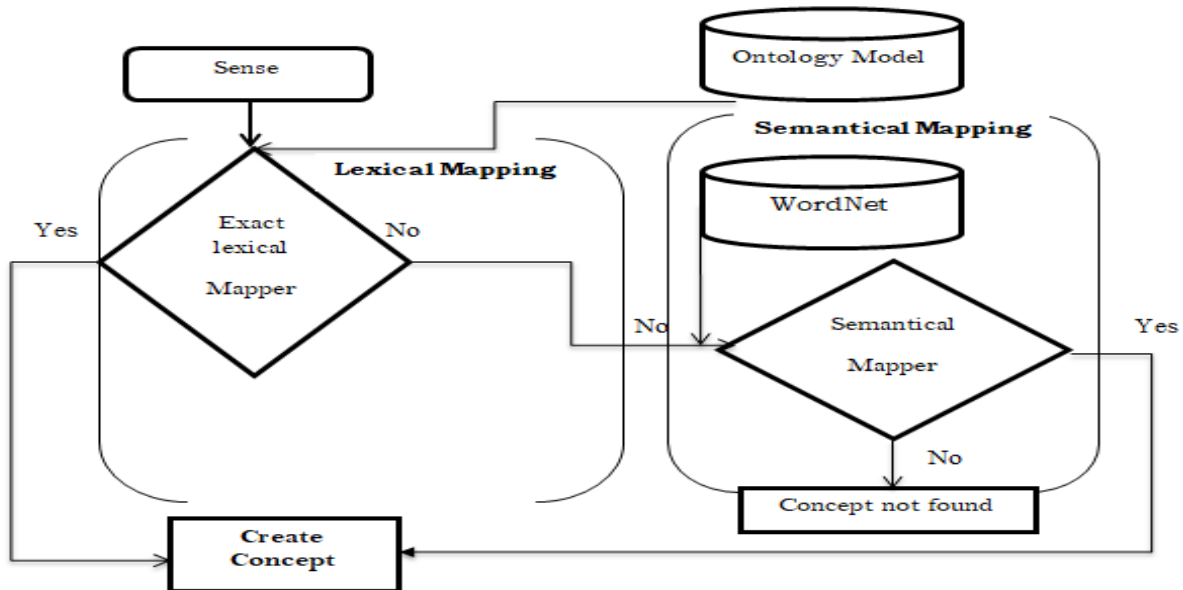


Figure 4.16: Sense to ontology concept mapping

Firstly, a sense generated from translation module is used for exact matching with ontology concept after stopword removal and stemming. If a sense has an exact match to any concepts in the ontology, mapped concept information is stored. If concept is not found with exact matching, try to perform an approximate string match method for both parties and save mapped concept information. After all this if match is not found, finally the system use an external lexical knowledge i.e. WordNet to perform a semantical matching and store the mapped concepts.

In addition to this, during concept mapping when compound words (or multiple words as single) are found, the mapping is processed word by word along both parties (i.e. ontology concepts and translated senses). In this investigation the compound words are separated with “_” and during mapping the compound word is split into sub words in order to map word by word using both lexical as well as semantical mapping techniques. Beside this, like that of mapping a single word, a pre-processing strategy (i.e. stopword removal and stemming) is also taken in each sub word of a compound word. This enhances the finding of a matching probability of concepts in both parties (i.e. translated senses and ontology concepts).

On the other hand, like that of text translation, mapping the ontology concept along with all translated senses is also repeated and time consuming process. Hence, in this investigation in order to minimize such inefficiency the lexical and semantical matching of each translated sense is done only once even if appears more than one times.

4.7 Classification

It is final module of the proposed multilingual text classifier and it is devoted to accept the bag of weighted concepts and assigns one or more concepts based on their importance as a category of a given document. It has only one component called text categorization. In this section we will briefly discuss how the text translation sub component operates.

4.7.1 Text Categorization

When the each translated senses are mapped along the ontology concepts, a weight is assigned to each concept based on mapped lemma occurrence in the document. So, in this phase, the ontology concepts are assigned weights in order to consider the frequency of the corresponding lemma in the document.

The process of weighting concepts according to the mapped translated sense is useful in order to discriminate the important and less important mapped concepts returned from the ontology. As explained before, the determinant that influences a weight given to a concept is the occurrence of a concept based on the number of lemma frequency. The lemma frequency indicates how frequent a particular concept is mentioned in the document. The higher the frequency, the more important the concept is considered to be.

In detail, each concept in ontology O is compared with all the translated senses of all the lists S_0 . Every time when a match is found, the label containing the weight of the concept is increased by the value associated with the list containing the matching lemma word.

$O_w = \text{assignWeights}(O)$ where O_w is the weighted ontology.

To assign weight to an ontology concepts C_i , which is mapped for a lemma L_j is the sum of lemma frequency of the j th lemma of a document d .

$$W_{ci} = \sum_{j=1}^n f(L_j) \quad (4.6)$$

Where W_{ci} refers to as weight of concept c_i , and $f(L_j)$ is frequency of lemma L_j

In this phase, all the concepts in O_w are visited and those with a weight greater than zero are inserted into the result list of weighted concepts.

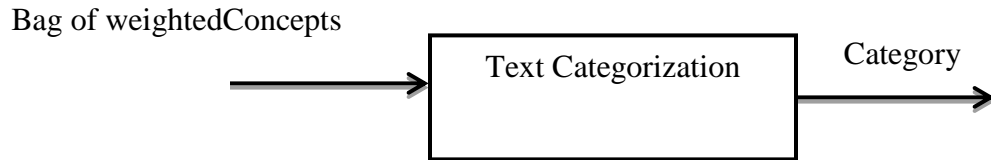


Figure 4.17: Text Categorization

Hence, after weighting each ontology concept, our classifier assigns one or more ontology concepts that have a maximum weight as a category of a given text document.

$$Category = \text{Max} \begin{cases} i = n \\ Ci \\ i = 1 \end{cases} \quad (4.9)$$

Where Category refers to assigned document category, C_i is ontology concepts having a weight of greater than zero.

Therefore, the document which contains the highest matching score with the corresponding top level ontology class is considered as the most suitable category of the given document.

4.8 Prototype

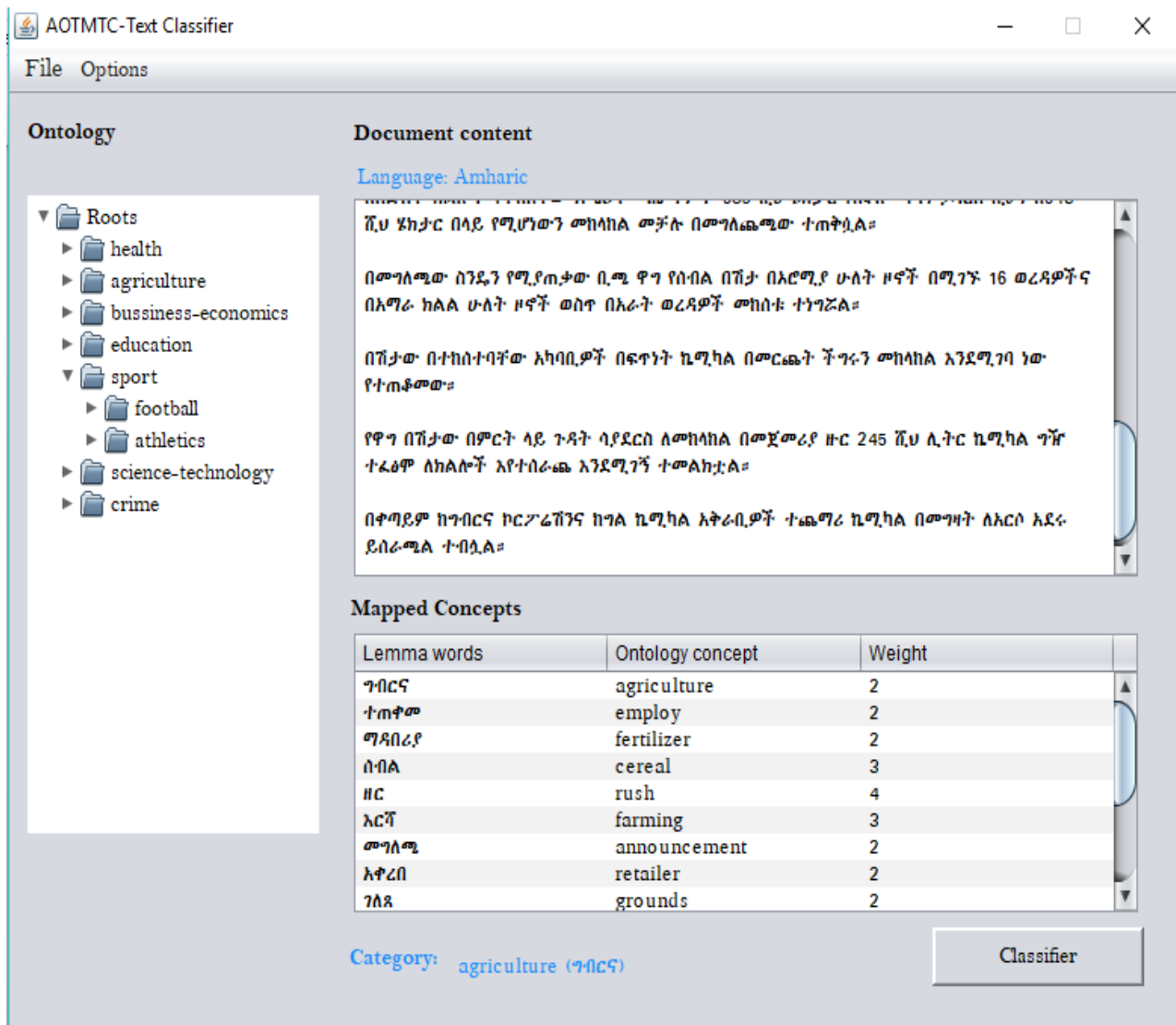


Figure 4.18: The screenshot of prototype's user

As shown in Figure 4.18 above, the Prototype interface presented information along four different output areas.

- **Ontology area:** - An area in which the ontology is loaded when the ontology is selected by the user with the file selection menu located on the top of Prototype interface. Beside this, the user can also navigate the ontology concepts by expanding the ontology tree.
- **Document content area:** - An area where the text content associated with the test document is loaded when file is selected by the user with the file selection menu located on the top of Prototype interface.

- **Mapped concepts area:-** An area in which the details of concept mapping output is presented. As shown in Figure 4.18, in this area lemma, associated ontology concept, and weight of ontology concept are displayed.
- **Category area:** - After the classification is computed with classification module, the final result i.e. one or more document categories are presented in this area. As shown in Figure 4.18 the final category result is presented in Ontology language as well as text document language.

4.9 Summary

In this chapter, we described the basic design criteria and elements of the framework for automatic multilingual under resourced textual documents with resourced language ontology. The main modules of the framework such as preprocessing module, text translation module, concept mapping module and classification module have been briefly presented. In addition to this, it also explained the about tools which were used as a component in our proposed framework.

This chapter also presented the design and implementation issues of each modules component of the framework briefly. As well, we have also discussed about the features of the proposed approach and also the way in which the prototype interface presents output information.

Chapter Five

Experiment

5.1 Introduction

Assigning document categories based on their content is not trivial, especially for multilingual textual documents which needs very complicated techniques and methods. To conduct the experiment, we have followed set of procedures which consist of set of activities. Preparing the corpus is one of the most important stages in this investigation. Hence, news related corpus is collected for the components of the proposed approach such as for POS tagging, language identification, normalization and so on. In the subsequent pages of this investigation, we will discuss the experimental procedures and the results.

5.2 Data Collection

In general, in this investigation we collect different types of documents related to News domain to achieve our ontology based multilingual text categorization task. Consequently, we have collected four types of corpus; it is classified based on the purpose in which the corpus is used in the development stage of the classifier. These are: corpus for language identification and stopword removal task, corpus to expand abbreviation of words, corpus for POS tagging task, corpus for bilingual dictionary, and news related documents for testing purpose.

(A). Corpus collected for language identification and stopword removal

As explained in chapter four, in order to perform the multilingual classification the language for the input document should be recognized or identified. A better approach chosen for this language identification task is stopword based approach. So, in order to achieve this, we adopted list of stopwords used by different researchers for each supported language. The Amharic stopwords used in this study is adopted from Eyob Delele works [92]. Similarly for Afan Oromo stopword lists, it is adopted from works of Fiseha Berhanu [93]. Furthermore, Tigrinya stopwords is adopted from Yonas Fisseha works [94]. For more clarification, stopword for Amharic, Afaan Oromo, and Tigrinya language used in this investigation is available in Appendix A.

As showed in chapter four figures 1.4, after the stopwords are used for language identification task, they are again used to identify words which are not important for document representation

in order to perform text categorization task. So, this list of stopwords used in language identification also used later for stopword removal operation as stated in chapter four.

(B). Corpus to expand abbreviation of words

As explained in chapter four, there is a module called normalization which dedicated to expand the abbreviated words into their longest form. In order to accomplish this we collect a list of abbreviated words with their corresponding longest form value for each supported language of the proposed multilingual text categorization. The list of abbreviated words with their normalized value used in this investigation is adopted from different previous research works. The Amharic list of this abbreviated words information is adopted from [101]. On the other hand, for Afaan Oromo list is information adopted from Fiseha Berhanu works on [93] and finally the Tigrinya list is adopted from [101].

(C). Corpus collected for POS tagging

As explained in chapter four, a part of speech tagging is used as a component to achieve our investigation and this is done by adopting the Java wrapper tool called TreeTagger. Consequently, to train this tool, we manually collected tagged training corpus for each supported language (i.e. Amharic, Afaan Oromo and Tigrinya). The training corpus is collected from HaBit (Harvesting big text data for under-resourced languages) [79], which is developed to gather large scale text data (corpora) from web for under-resourced languages.

The size of tagged corpus taken from HaBit project to train the treetagger is described under Table 5.1. The TreeTagger is adaptable to other languages if a lexicon and a tagged corpus are available [80].

Language	Word	Sentence	Tag
Amharic	17,320,000	1,208,926	33
Afaan Oromo	4,249,953	250,432	12
Tigrinya	2,087,613	139,357	15

Table 5.1: Treetagger training corpus size statistics

(D). Corpus to build bilingual dictionary

Bilingual dictionary (i.e. Amharic to English, Afaan Oromo to English and Tigrinya to English) is the main resource for lexical knowledge of the translation module. The bilingual dictionary provides a target textual meaning or synonyms of the source language words in order to make our classifier to operate along the multilingual environment. To construct our bilingual dictionary for the supported language, we collected dictionary based corpus from different sources.

In order to construct Amharic to English bilingual dictionary, we used a google translator, which is a free multilingual machine translation service developed by [Google](#) [95]. In addition to this, we also used different hardcopy dictionary such as Amharic to English bilingual dictionary compiled by Endale Zenawi [97].

To build Afaanoromo to English bilingual dictionary, we used an Afaanoromo-English bilingual dictionary [96]. In order to make this dictionary suitable for our work, we perform preprocessing tasks such as converting the image scanned dictionary to content editable softcopy format, arrange the word list structure to be suitable for our investigation and so on.

On the other hand, in order to compile and build Tigrinya to English bilingual dictionary, we used an online free distributed Tigrinya to English bilingual dictionary compiled by Efreem Zecarias [98].

The vocabulary size of each bilingual dictionary of the supported language used for this investigation is described in the Table 5.2.

	Amharic – English	Afaan Oromo – English	Tigrinya – English
Vocabulary size	23,917	7,020	9,841

Table 5.2: Bilingual dictionary size statistics

(E). Corpus collected for testing purpose

In order to perform our experiment and evaluate the performance of the proposed approach, the test documents are chosen from each of the news domain specified category used in this investigation such as Politics, Business and Economy, Sport, Health, Education and Science and Technology. Then all documents are used to evaluate our system see under the experiment section of this investigation. These testing documents are collected from different sites for all supported language of the proposed classifier.

The Amharic test document for the specified news domain category is collected from Fana Broadcasting Amharic program [102] and VOA (Voice of America) [104]. In addition, the Afaan Oromo test document for the specified news domain category is also collected from Fana Broadcast Afaan Oromo program [102]. On the other hand, for Tigrinya test document for the specified news domain category is collected for dimtsi weyane [103] and VOA [105].

On the other hand, in order to evaluate the proposed approach for Amharic and Tigrigna 20 documents are used in each news category. As well, for Afaan Oromo 15 documents in each news category are used.

5.3 Implementation

The development tool selected was pure object oriented programming, particularly Java programming language. Hence, among different benefits of OOP in comparison of other system development that is easy: to develop, manipulate, test and understand. Because, OOP clusters things in terms of class and objects so, the procedure to undertake by accessing or not to accessing different module according to the given experimentation techniques.

For example: experiment 2 was conducted by the procedure of not to access preprocessing module (i.e. stopword removal and stemming) during concept mapping. In short, the procedure used in all experiment is almost the same; the only difference is the class they access.

5.4 Evaluation

To evaluate the proposed multilingual under-resourced document classifier, first the test data corpus is prepared by gathering different news category documents. After those news categories documents gathered from different news portal, the next step is labeling one or more news categories to these documents manually for testing purpose. In order to approve manually classified categories and sub-categories of test documents, domain experts were involved. The manually classified documents help for checking the final result of the automatic document categorizer.

5.4.1 Evaluation Metrics

Evaluation of the classifier is done with the evaluation parameter that compares the number of documents which are classified correctly and incorrectly. Typically, the comparison is done amid

the document classified using the automatic classifier and that of the manually classified documents.

As mentioned in chapter 2, the evaluation parameters used in this investigation are recall, precision and F-measure, details description of these evaluation metrics is mentioned in chapter 2.

On the other hand, choosing index terms with n number of frequency depends on the condition; it is possible to take the value of n from one up to the maximum number of frequency. Particularly in this investigation, this decision made on the frequency of index terms is based on the maximum number of documents to be categorized. Hence, in this investigation for either with or without morphological analysis, index terms are selected with frequency greater than or equal to two achieve a better result of document categorization. Therefore, throughout the experiment index terms are selected starting from two up to the maximum frequency.

5.4.2 Test Result

In this investigation four experimental objectives were proposed to observe the strength of proposed multilingual classifier for under-resourced documents from different perspectives. In general in this thesis the following four proposed experiments were undertaken.

In this section in order to make suitable our table based description of the experimental result we abbreviated Amharic, Afaan Oromo and Tigrigna as A, O and T respectively.

Experiment 1: examining the performance of proposed text classifier

The objective of this experiment is to discuss the performance of the proposed text classifier when all features and components are incorporated. Table 5.3 shows the confusion matrix which summarizes the evaluation results of experiment 1 for all supported language documents (i.e. Amharic, Afaan Oromo and Tigrigna) along with eight news category.

The confusion matrix depicts documents which are wrongly classified for each news categories along the supported language (i.e. Amharic, Afaan Oromo and Tigrinya).

	C1 Agriculture			C2 Business and economics			C3 Crime			C4 Education			C5 Health			C6 Science and technology			C7 Athletics			C8 Football		
	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T
C1	17	13	18	2	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	
C2	0	1	1	20	13	18	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	
C3	0	0	0	0	0	0	19	13	18	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0
C4	0	1	0	0	0	0	0	0	0	19	12	17	0	0	1	1	2	2	0	0	0	0	0	0
C5	0	0	0	1	0	0	0	0	0	0	0	1	18	14	19	0	0	0	0	1	0	1	0	0
C6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	19	13	18	0	0	0	0	0	0
C7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	17	13	18	3	1	1
C8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	2	1	20	12	18

Table 5.3: Confusion matrix for experiment 1

Based on Table 5.3 confusion matrix of experiment 1, the accuracy of this experiment is evaluated. Hence, Table 5.4 shows the calculated values of these measures (i.e. Recall, Precision and F-measure) for the experiment 1.

News Category	A Amharic			O Afaan Oromo			T Tigrinya		
	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)
Agriculture	100	85	91.90	86.67	86.67	86.67	85.71	90	87.80
Business and economics	86.96	100	93.03	92.86	86.67	89.66	94.74	90	92.31
Crime	100	95	97.43	100	86.67	92.86	100	90	94.74
Education	100	95	97.43	92.31	80	85.71	89.47	85	87.18
Health	90	90	90	70	93.33	79.99	76	82.61	79.17
Science and technology	90	95	92.43	86.67	86.67	86.67	85.71	85.71	85.71
Athletics	100	85	91.90	76.47	86.67	81.25	81.82	90	85.72
Football	80	100	88.8	92.31	80	85.72	94.74	90	92.31
Average (%)	93.37	93.13	92.87	87.17	85.84	86.07	88.52	87.92	88.12

Table 5.4: Precision, Recall and F-measure results for experiment 1

Experiment 2: examining the effect of morphological analyzer during index term selection

The major intention of this experiment is to measure the performance of the proposed text classifier using only original terms occurrence based index term selection. In other words, to observe the significance of Morphological Analysis during index term selection and also to observe the capability of pure term frequency based index term selection of this investigation.

Below Table 5.5 depicts the confusion matrix which summarizes the results of testing experiment 2. Like experiment 1, the experiment is done for all supported language (i.e. Amharic, Afaan Oromo and Tigrinya) and with eight news categories documents.

	C1 Agriculture			C2 Business and economics			C3 Crime			C4 Education			C5 Health			C6 Science and technology			C7 Athletics			C8 Football		
	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T
C1	16	12	15	2	1	2	0	0	0	0	0	1	2	2	1	0	0	0	0	0	1	0	0	0
C2	1	1	2	18	13	16	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
C3	0	0	0	0	0	2	19	12	17	0	0	0	0	2	1	0	1	0	0	2	0	1	0	0
C4	0	1	0	1	2	1	0	0	0	18	11	16	0	0	1	1	1	2	0	0	0	0	0	0
C5	0	1	2	1	0	0	0	0	0	0	0	1	16	12	17	0	0	0	1	2	0	2	0	0
C6	0	1	0	0	0	0	0	0	0	1	1	2	1	1	1	18	12	17	0	0	0	0	0	0
C7	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	17	12	16	4	1	2
C8	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	2	1	19	12	17

Table 5.5: Confusion matrix for experiment 2

Once the confusion matrix is computed as presented in Table 5.5, we can compute the evaluation measures (i.e. Recall, Precision and F-measure) of experiment 2. Hence, Table 5.6 illustrates the experimental result of experiment 2 based on evaluations of these measures.

News Category	A Amharic			O Afaan Oromo			T Tigrinya		
	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)
Agriculture	94.12	80	86.5	75	80	77.42	78.95	75	76.92
Business and economics	81.82	90	85.72	76.47	86.67	81.25	76.19	80	78.04
Crime	100	95	97.43	100	80	88.89	100	85	91.89
Education	90	90	90	91	73.33	81.21	72.72	80	76.19
Health	76.19	80	78.05	63.16	80	70.59	73.91	85	79.07
Science and technology	94.74	90	92.30	92.30	80	85.71	85	85	85
Athletics	93.75	75	83.33	66.67	80	72.72	90	90	90
Football	73.08	95	82.61	92.30	80	85.71	94.44	85	89.47
Average (%)	87.96	86.88	86.99	82.11	80	80.38	83.90	83.12	83.32

Table 5.6: Precision, Recall and F-measure results for experiment 2

Experiment 3: examining the effect of stopword removal and stemming during concept mapping

The objective of this experiment is to show the performance of proposed multilingual under-resourced document classifier without pre-processing modules (i.e. stopword remover and stemmer) during concept mapping. Conversely, to observe the capability of concept mapper when stopword remover and stemmer components are not incorporated.

Hence, Table 5.7 presents the confusion matrix of this experiment for all supported languages along eight news categories documents.

	C1 Agriculture			C2 Business and economics			C3 Crime			C4 Education			C5 Health			C6 Science and technology			C7 Athletics			C8 Football		
	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T
C1	14	11	15	2	2	2	0	0	0	0	0	1	3	2	1	1	0	1	0	0	0	0	0	
C2	2	2	2	16	10	16	0	0	0	0	1	1	0	2	0	1	0	1	1	0	0	0	0	
C3	0	0	0	1	0	1	16	11	15	0	0	0	1	1	2	1	2	1	0	1	1	1	0	
C4	1	1	0	0	1	0	0	0	1	17	12	15	1	1	1	1	0	2	0	0	1	0	0	
C5	2	1	1	1	0	1	0	0	0	1	1	1	15	12	17	0	0	0	0	1	0	1	0	
C6	1	0	0	0	1	2	0	0	0	2	2	1	1	1	1	16	11	16	0	0	0	0	0	
C7	0	0	0	0	0	0	0	0	0	0	0	0	2	2	1	0	0	0	15	12	17	3	1	
C8	0	1	0	1	0	0	0	0	0	0	0	1	1	1	2	0	0	0	2	2	1	16	11	

Table 5.7: Confusion matrix for experiment 3

Once the confusion matrix of experiment 3 is computed as presented in Table 5.7, we compute the accuracy of the proposed approach through Precision, Recall and F-measure and the result is illustrated in Table 5.8.

News Category	A Amharic			O Afaan Oromo			T Tigrinya		
	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)
Agriculture	70	70	70	68.75	73.33	70.97	83.33	75	78.95
Business and economics	76.19	80	78.05	71.42	66.67	68.97	72.73	80	76.19
Crime	100	80	88.89	100	73.33	84.61	93.75	75	83.33
Education	85	85	85	75	80	77.41	75	75	75
Health	62.5	75	68.18	54.55	80	64.87	68	85	75.56
Science and technology	80	80	80	84.62	73.33	78.57	76.19	80	78.05
Athletics	83.33	75	78.95	75	80	77.42	85	85	85
Football	76.19	80	78.05	91.67	73.33	81.48	88.89	80	84.21
Average (%)	79.15	78.13	78.38	77.63	74.88	75.54	80.25	79.38	79.54

Table 5.8: Precision, Recall and F-measure results for experiment 3

Experiment 4: examining the effect of semantics based matching during concept mapping

The major intention of this experiment is to measure the performance of proposed text classifier when semantics consideration between ontology concept and translated sense is not incorporated during concept mapping. Hence, in order to measure the effect of this experiment object, we compute a confusion matrix as presented in Table 5.9. The confusion matrix illustrates the correctly as well as wrongly classified documents in each news category for all supported languages (i.e. Amharic, Afaan Oromo and Tigrinya). Like pervious experiments, for evaluation purpose we used documents for each news category.

	C1 Agriculture			C2 Business and economics			C3 Crime			C4 Education			C5 Health			C6 Science and technology			C7 Athletics			C8 Football		
	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T	A	O	T
C1	16	11	15	2	2	2	0	0	0	0	1	0	2	1	2	0	0	1	0	0	0	0	0	0
C2	0	2	1	17	10	16	0	0	1	0	0	0	1	2	1	2	1	1	0	0	0	0	0	0
C3	0	0	0	1	1	1	16	9	15	0	0	0	1	2	1	1	2	2	0	1	1	1	0	0
C4	0	2	1	2	1	0	0	0	0	16	10	15	0	0	2	2	2	2	0	0	0	0	0	0
C5	2	0	1	1	0	0	0	0	0	1	0	2	14	12	16	1	2	1	0	1	0	1	0	0
C6	0	2	1	0	0	0	1	0	1	1	1	1	1	2	2	17	10	15	0	0	0	0	0	0
C7	0	0	0	0	0	1	0	0	0	0	1	0	2	2	1	0	0	0	15	11	16	3	1	2
C8	0	0	0	0	0	1	0	0	0	1	1	0	1	2	2	0	0	0	2	2	2	16	10	15

Table 5.9: Confusion matrix for experiment 4

In this experiment, in order to clearly identify the best alignment between the ontology concept and translated sense or to check the similarity between these two party components a confidence level value is required after the computation. We adopt a threshold value as similarity identifier value from works on [119] and the ontology concepts and translated sense are considered as semantically related entities when at least their hybrid computation of these three methods achieves 0.80. Hence, throughout this experiment this threshold value is used.

From the resulting confusion matrix of experiment 4, as illustrated in Table 5.9, we can compute the proposed text classifier accuracy based on Precision , Recall and F-measure and the computed result of these measurements for this experiment is depict in Table 5.10.

News Category	A Amharic			O Afaan Oromo			T Tigrinya		
	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)	R (%)	P (%)	F-m (%)
Agriculture	88.89	80	84.21	64.71	73.33	68.75	78.95	75	76.92
Business and economics	73.91	85	79.07	71.43	66.67	68.97	76.19	80	78.05
Crime	94.12	80	86.49	100	60	75	88.24	75	81.08
Education	84.21	80	82.05	71.43	66.67	68.97	83.33	75	78.95
Health	63.64	70	66.67	52.17	80	63.16	59.26	80	68.09
Science and technology	73.91	85	79.07	58.82	66.67	62.5	68.18	75	71.43
Athletics	88.24	75	81.08	73.33	73.33	73.33	84.21	80	82.05
Football	76.19	80	78.05	90.91	66.67	76.93	88.24	75	81.08
Average (%)	80.39	79.38	79.59	72.75	69.17	69.70	78.33	76.88	77.21

Table 5.10: Precision, Recall and F-measure results for experiment 4

5.4.3 Discussion

As shown in Table 5.4, when all features are incorporated, the proposed multilingual under-resourced document classifier achieved average F-measure of 92.37%, 86.07% and 88.12% for Amharic, Afaan Oromo and Tigrinya documents respectively. The result form this experiment shows very promising result (Table 5.4 above). The result of experiment implies directly the strength and efficiency of the proposed approach with all combined features used in this investigation. However, the performance decreased for Afaan Oromo as well Tigrinya documents and this is due to the small vocabulary size of compiled bilingual dictionary used for both languages.

On the other hand, from the result of experiment 2 that is shown in Table 5.6, the performance of the proposed approach is decreases with 5.38 %, 5.19% and 4.8% for Amharic, Afaan Oromo and Tigrigna documents respectively when lemma based index term selection is not incorporated. This is because of that words which are equivalent but inflected are treated as different words during index term selection. Hence, documents are not indexed with accurate

terms or terms which need to represent a document are reduced with term frequency threshold value. Since, applying better dimensional reduction or feature selection technique is beneficiary for increasing reliability and accuracy of text classification.

Furthermore, from the result of the experiment 3 shown in Table 5.8 the performance of the system decreases when the stopword remover and stemmer are not incorporated in the text categorizer. This is because the concept mapping cannot match par or compound words which are vary based on the constituent stopwords in both ontology concept and translated sense. Such constituent of stopwords decreases the probability of finding an exact lexical matching during concept mapping module. For instance, a word “meeting of place” and “meeting place” are considered as different words during extract lexical concept mapping and this reduces the text categorization performance as whole. On the other hand, the ontology concepts and translated sense which have same root word but inflect words can be considered as different words during exact lexical matching. Due to this the probability of matching such inflected words in both parties is reduced and the performance of document categorization as whole decrease. However, the involvement of stemmer in both ontology concepts and translated sense increase an exact lexical matching of both parties. Finally, due to noninvolvement of such pre-processing modules (i.e. stopword removal and stemmer) during concept mapping the performance of the proposed document classifier degrades with an average F-measure of 13.99%, 10.53%, and 8.58 % for Amharic, Afaan Oromo and Tigrinya respectively.

As we illustrated in Table 5.10, the evaluation results of experiment 4, the proposed multilingual under-resourced document classifier performance reduces with average F-measure of 13.28%, 16.37% and 10.91% for Amharic, Afaan Oromo and Tigrigna documents respectively. Hence, from this experiment we observed that lack of semantics based concept mapping highly affects the performance of whole system. Since, either lexical matching cannot check whether the ontology concept and translated sense are semantically related or not. There are cases where lexical metrics fail to identify similarity between strings that are terminologically different.

Chapter Six

Conclusion and Recommendations

6.1 Conclusion

The explosion of the World Wide Web provides a growing amount of information and data coming from different sources. Therefore, a text categorization mechanism is required for finding, filtering and managing the rapid growth of online information.

As explained earlier, based on a number of previous researchers analysis, ontology based text categorization approach outperforms the machine learning or keyword based text categorization approach. However, it is very challenging to build ontology of under resourced language (i.e. Amharic, Afaan Oromo and Tigrinya) from the scratch. Since, building domain ontology needs domain knowledge as well as language resource.

So, in order to eliminate this difficulty , this investigation designed and developed a text classifier which was able to categorized textual documents written with under-resourced languages (i.e. Amharic, Afaan Oromo and Tigrinya) on top of resourced language ontology (i.e. English ontology). Our approach consists of several modules: pre-processing module, text translation module, concept mapping module and classification module. As explained earlier, these higher level modules of our approach consists of several subcomponents that were discussed earlier, which made this investigation achievable. Finally, we conducted four experiments and evaluated our approach based on basic evaluation metrics: precision, recall and F-measure. The evaluation result of the proposed text classifier show that the proposed approach with incorporation of all features and components achieved a better result , an average F-measure of 92.37%, 86.07% and 88.12% for Amharic, Afaan Oromo and Tigrigna document. This experimental result indicated that the proposed approach was able to classify documents effectively when all features and components (i.e. lemma based index term selection, pre-processing strategies (i.e. stopword removal and stemming) during concept mapping and semantical based concept mapping) were incorporated.

In this investigation, the number of supported under resourced language to be classified is limited to Amharic, Afaan Oromo and Tigrinya. However, thanks to the system modularity it is

possible to extend for other languages if bilingual dictionary, TreeTagger training corpus and other language specific lexicons are available. As well, in this investigation for experimental purpose we adopt English ontology in the news domain, but also due to system flexibility and modularity it can be used other domain ontology.

6.2 Contribution of the study

Some of the main contributions of the study are listed below:

- A generic model is proposed for multilingual under-resourced document categorization on top of resourced ontology. Hence, the study shows the possibility of achieving classification of multilingual under-resourced documents using English ontology.
- Enhancement of index term selection through the lemma occurrence in actual document instead of the actual term itself. This solves the problem of considering inflected words as different during term frequency computation and enhances the text categorization process.
- Using pre-processing strategies (i.e. stopword removal and stemming) during mapping of ontology concepts along with translated senses. This enhances the probability of finding a match in both parties.
- Using combination of semantic relatedness measurements along the ontology concepts and translated senses based on WordNet: edge based similarity measure (i.e. shortest Path based Measure, Wu & Palmer's Measure) and information content based similarity measure (i.e. Lin's Measure).
- In addition, the study contributes to the growth of semantics technology as well as text categorization. Since, the proposed multilingual under-resourced document categorization paves the way for text categorization with semantic technologies.

6.3 Recommendations

The results found in this research showed that classification can be done automatically for under-resourced language documents using resourced language ontology. However, to enhance more the quality and performance of the multilingual text classifier, the following ideas are recommended for further research work.

- In this investigation, the bilingual dictionary used as knowledge source for the text translation module was built on a small size of vocabulary terms; however, for further study and to gain better performance of generating a sense of target language during word by word translation it is recommended to enhance a well-qualified vocabulary terms of bilingual dictionary for all supported languages of text classifier.
- In this investigation, we adopt a TreeTagger using a supported language lexicon as well as manually tagged training corpus. The performance of TreeTagger depends on the size and quality of training corpus used. As well, TreeTagger was used as a core component during text translation module and lemma identification for disambiguation purpose based on the tag of a word. So, in order to optimizing the search within the bilingual dictionary it is recommended for the improvement of the training corpus in terms of size as well quality.
- From the experiment that have been conducted , in this work absence of morphological analyzer degrades the performance of the text classifier by average F-measure of 5.38 % , 5.19% and 4.8% for Amharic , Afaan Oromo and Tigrinya documents. On the other hand, the presence of the morphological analyzer for index term selection increases the performance of text categorizer. Having this in mind, the presence of most powerful tool which can generate the root or lemma of a word, or a tool which can handle highly morphologically inflected word is recommended.
- In this paper, the evaluation was carried out on small data sets; however, for further study and to gain better performance, increasing the number of experiment can make the work more robust.

References

- [1] S. International, "Tigrinya," in Tigrinya at Ethnologue, Ethnologue, 2015. [Online]. Available: <http://www.ethnologue.com/18/language/tir/>. Accessed: Nov. 20, 2016.
- [2] A. SinghRathore and D. Roy, "Ontology based web Page Topic identification," International Journal of Computer Applications, vol. 85, no. 6, pp. 35–40, Jan. 2014.
- [3] L. Tenenboim, B. Shapira, and P. Shoval, "Ontology based classification of News in an Electronic Newspaper," Intelligent Information and Engineering Systems, Jun. 2008.
- [4] K. Taghva, J. Borsack, J. Coombs, A. Condit, S. Lumos and T. Nartker, "Ontology based classification of Email", in International Conference on Information Technology, 2003.
- [5] H. Dong, E. Chang and F. Hussain, "An Ontology based Webpage Classification Approach for the Knowledge Grid", in Fifth International Conference ON Semantics, Knowledge, 2009.
- [6] P. Carol and S. Peters. Accès multilingue aux systèmes d’information. In 67th IFLA Council and General Conference, 2001.
- [7] N. Geoffrey. Will the Internet Always Speak English. The American Prospect, 11(10), 2000.
- [8] M. Sahlemariam, M. Libsie and D. Yacob, "Concept-Based Automatic Amharic Document Categorization", in Americas Conference on Information Systems (AMCIS), 2009.
- [9] B. Smith, "Beyond Concepts: Ontology as Reality Representation", in International Conference on Formal Ontology and Information Systems, 2004.
- [10] D. Nichols and A. Terry, "User's Guide to Teknowledge Ontologies" Teknowledge Corp. December 3, 2003
- [11] T. T. D. Group, "About world languages," in OWL about world languages, 2015. [Online]. Available: <http://aboutworldlanguages.com/>. Accessed: Dec. 10, 2016.
- [12] L. Wolf, Documents Tigrinya. Paris: Libraire CKlincksieck., 1998.
- [13] G. Assefa, A two-step approach for Tigrinya text categorization. MSC Thesis. Addis Ababa University, Addis Ababa, Ethiopia, 2011.
- [14] D. Oard and B. Dorr, A survey of multilingual text retrieval. Technical report, College Park, MD, USA, 1996.

- [15] G. De Melo and S. Siersdorfer, Multilingual text classification using ontologies. In: *Advances in Information Retrieval*. Springer, 2007.
- [16] S. Vogrincic, and Z. Bosnic, "Ontology-based multi-label classification of economic articles." *Comput. Sci. Inf. Syst.*, 8(1), 101-119, 2011.
- [17] L. David, Feature selection and feature extraction for text categorization. In: *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 212-217, 1992.
- [18] A. Segev. & A. Gal, Enhance Portability with Multilingual Ontology Based Knowledge Management. Technion: Israel Institute of Technology. Isreal, 2008.
- [19] A. Ferrando, S. Beux & V. Mascardi, MOo-TC, a Multilingual Ontology Driven Text Classifier. University of Genova, Italy, 2015.
- [20] T. Goncalves & P. Quaresma, Multilingual Text Classification through combination of monolingual classifiers. University: Evora, 2010.
- [21] A. Addis, "Study and Development of Novel Techniques for Hierarchical Text Categorization", *University of Cagliari, Italy*, 2010.
- [22] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney, "Feature selection methods for text classification", In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, California, USA, August 12-15, 2007, pages 230–239, 2007.
- [23] A. Ozgur, "Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization", *MSc Thesis. Bogazici University, Turkey*, 2004.
- [24] A. Y. Ng, M. I. Jordan, "On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes", *NIPS*. pp. 841- 848, 2001.
- [25] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification", In *AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48, AAAI Press, 1998.
- [26] A. Bouiadra , S. Bensliman , "FOEval: Full ontology evaluation", 7th International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), pp.464, 468, 27-29 November, 2011.
- [27] A. Kayed , "Ontology evaluation: Which test to use", 5th International Conference on Computer Science and Information Technology (CSIT), 2013, pp.45,48, 27-28 March 2013.

- [28] C. Shweta , D. Maya , I. P. Kulkarni, “A Comparative Analysis of Supervised Multi-label Text Classification Methods”, International Journal of Engineering Research and Applications (IJERA) Vol. 1, Issue 4, Page .No 1952-1961, March 2012.
- [29] C. Ming-Syan , H. Jiawei Han, P.S. Yu , "Data mining: an overview from a database perspective," in Knowledge and Data Engineering, IEEE Transactions on , vol.8, no.6, pp.866-883, Dec 1996.
- [30] C. Stokoe, “Differentiating homonymy and polysemy in information retrieval”: In HLT/EMNLP, The Association for Computational Linguistics, 2005.
- [31] D. D. Lewis and K. A. Knowles, “Threading electronic mail - A preliminary study”: Inf. Process. Manage, 33(2):209–217, 1997.
- [32] D. H. Fusilier, M. M. y Gómez, P. Rosso, and R. G. Cabrera, “Detecting positive and negative deceptive opinions using pu-learning”, Information Processing & Management, 51(4):433 – 443, 2015.
- [33] D. Vickrey, L. Biewald, M. Teyssier, and D. Koller, “Word-sense disambiguation for machine translation”, In Conference on Empirical Methods in Natural Language Processing (EMNLP), Vancouver, Canada, October 2005.
- [34] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades , “Ontologybased sentiment analysis of twitter posts”, Expert Syst. Appl., 40(10):4065–4074, 2013.
- [35] E. W. De Luca, A. Nrnberger, “Using clustering methods to improve ontology-based query term disambiguation”, International Journal of Intelligent Systems, 21:7, pp. 693-709, 2006.
- [36] E. Alatrash, “Using Web Tools for Constructing an Ontology of Different Natural Languages”, Doctoral dissertation, University of Belgrade, 2013.
- [37] E. W. De Luca, A. Nrnberger, “Ontology-based semantic online classification of documents: Supporting users in searching the web”, Proc European Symp on Intelligent Technologies, Aachen, 2004.
- [38]F. Sebastiani, "Text categorization. In Text Mining and its Applications to Intelligence, CRM and Knowledge Management", WIT Press, pp. 109–129, 2005.
- [39] F. Sebastiani ,“Text Categorization”,Dipartimento di Matematica Pura e Applicata, Universit`a di Padova, 35131 Padova, Italy.
- [40] F.Sebastiani, “Machine learning in automated text classification”, ACM Computing Surveys, vol. 34, no. 1, pp. 1-47,2002.

- [41] G. Assefa, "A two-step approach for Tigrinya text categorization", MSC Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2011.
- [42] G. Salton and M. McGill, *Introduction to modern information retrieval*, 1st ed. Auckland [u.a.]: McGraw-Hill Intern., 1987.
- [43] H. Berger, "A Comparison of Text Categorization Methods Applied to N-Gram Frequency Statistics", in *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, Australia, 2004, pp. 4-10.
- [44] I. Sandu Popa, K. Zeitouni, G. Gardarin, D. Nakache and E. Métais, "Text Categorization for Multi-Label Documents and Many Categories", *Washington DC, USA: IEEE*, 2007.
- [45] J. Ma, W. Xu, Y. Sun, E. Turban, S. Wang, O. Liu, "An Ontology-Based Text-Mining Method to Cluster Proposals for Research Project Selection" *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.42, no.3, pp.784,790, May 2012.
- [46] K. Taghva, J. Borsack, J. Coombs, A. Condit, S. Lumos and T. Nartker, "Ontology based classification of Email", in *International Conference on Information Technology*, 2003.
- [47] K. Lang. Newsweeder, "Learning to filter netnews", In *Proceedings of the 12th International Machine Learning Conference (ML95)*, 1995.
- [48] K. Bhaskar, S. Savita, "A Comparative Study of Ontology building Tools in Semantic Web Applications", *International journal of Web & Semantic Technology (IJWesT)* Vol.1, No.3, July 2010.
- [49] L. Tenenboim, B. Shapira, and P. Shoval, "Ontology based classification of News in an Electronic Newspaper," *Intelligent Information and Engineering Systems*, Jun. 2008.
- [50] L. S. Larkey, "A Patent Search and Classification System", *New York, USA: ACM*, 1999.
- [51] L. Seremeti, A. Kameas, "A Task-Based Ontology Engineering Approach for Novice Ontology Developers", *Fourth Balkan Conference in Informatics*, 2009. BCI '09, pp.85, 89, 17-19 September 2009.
- [52] M. Rada, B. Carmen and W. Janyce, "Learning multilingual subjective language via cross lingual projections", *ACL*, pp. 976–983, 2007.
- [53] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk E-mail", In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

- [54] M. Antonie and O. Zaiane, "Text Document Categorization by Term Association", in *Proceedings of the 2002 IEEE International Conference on Data Mining*, Japan, 2002, pp. 30-50.
- [55] M. Saad and W. Ashour, "Arabic Text Classification Using Decision Trees", In *Proceedings of the 12th international workshop on computer science and information technologies CSIT*, pages 75–79, 2010.
- [56] M. Bijaksana, Y. Li, and A. Algarni, "Scoring-thresholding pattern based text classifier", In A. Selamat, N. Nguyen, and H. Haron, editors, *Intelligent Information and Database Systems*, volume 7802 of *Lecture Notes in Computer Science*, pages 206–215. Springer Berlin Heidelberg, 2013.
- [57] M. Fernández-López, A. Gómez-Pérez, J. Euzenat, A. Gangemi, Y. Kalfoglou, D. Pisanelli, & Y. Sure, "A survey on methodologies for developing, maintaining, integrating, evaluating and reengineering ontologies", *OntoWeb deliverable D*, 1, 2002.
- [58] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries": how to tell a pine cone from an ice cream cone, In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, NY, USA, 1986. ACM.
- [59] O. Corcho, M. Fernández-López, & A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?", *Data & knowledge engineering* 46, pp. 41-64, 2003.
- [60] P. Edmonds and E. Agirre. "Word sense disambiguation". 3(7):4358, 2008.
- [61] R. Navigli and M. Lapata, "An experimental study of graph connectivity for unsupervised word sense disambiguation", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):678–692, April 2010.
- [62] S. Teklu, "Automatic categorization of Amharic news document: A machine learning Approach", *Master thesis, Addis Ababa University*, 2003.
- [63] "Semi-supervised learning literature survey", 2017. [Online]. Available: <http://pages.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html>. [Accessed: 18- Mar- 2017].
- [64] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features", In *Proceedings of the 10th European Conference on Machine Learning, ECML '98*, pages 137–142, London, UK, UK, 1998. Springer-Verlag.
- [65] T. R. Gruber, "A translation approach to portable ontology specifications", *Knowl. Acquis.*, 5(2):199–220, June 1993.
- [66] "Understanding Knowledge Societies", In *twenty questions and answers with the Index of Knowledge Societies*, United Nations New York, 2005.

- [67] V. R. Carvalho and W. W. Cohen, "On the collective classification of email "speech acts"," in SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005, pp. 345–352.
- [68] W. Cohen, "Learning rules that classify e-mail", In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning and Information Access, 1996.
- [69] Y. Afework, "Automatic Amharic Document Categorization": the case of Ethiopian News Agency", *Master thesis Addis Ababa University*, 2007.
- [70] Y. Yang, "An evaluation of statistical approaches to text categorization", *Journal of Information Retrieval*, 1:67–88, 1999.
- [71] Y. Liu; Z. Shao, "A framework for semantic Web Services annotation and discovery based on ontology", 2010 IEEE International Conference on Progress in Informatics and Computing (PIC), vol.2, pp.1034,1039, 10-12 December 2010.
- [72] Y. Chi Kiong , S. Palaniappan, N. A. Yahaya , "Health ontology system", 2011 7th International Conference on Information Technology in Asia (CITA 11), vol., no., pp.1, 4, 12-13 July 2011.
- [73] Y. Sure, S. Staab, & R. Studer, "Handbook on ontologies", Springer Berlin Heidelberg, pp. 135,152, 2009.
- [74] A. Kumilachew, "Hierarchical Amharic News Text Classification", MSc Thesis. Addis Ababa University, Addis Ababa, Ethiopia., 2010.
- [75] G. Wei, G. Wu, Y. Gu and Y. Ling, "An Ontology Based Approach for Chinese Web Texts Classification", *Information Technology Journal*, vol. 7, no. 5, pp. 796-801, 2008.
- [76] J. Ma, W. Xu, Y. Sun, E. Turban, S. Wang and O. Liu, "An Ontology-Based Text-Mining Method to Cluster Proposals for Research Project Selection", *IEEE Transactions on Systems*,
- [77] J. Chen, H. Huang, S. Tian and Y. Qu, "Feature selection for text classification with Naïve Bayes", *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432-5435, 2009.
- [78]. Truić a, C., Velcin, J. and Boicea, A. (2015). Automatic Language Identification for Romance Languages using Stop Words and Diacritics.
- [79] "HaBiT Project", *Habit-project.eu*, 2014. [Online]. Available: <http://habit-project.eu/>. [Accessed: 14- Jun- 2017].
- [80] Helmut S. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.
- [81] Solomon N. (2007). *Analysis of Semantic Technologies for Ethiopic Manuscripts, Art and Music*.
- [82] M. Gasser, "HORN MORPHO", 2017.

- [83]"Java Tutorial", www.tutorialspoint.com, 2017. [Online]. Available: <https://www.tutorialspoint.com/Java>. [Accessed: 4- July- 2017].
- [84] Levenshtein, Vladimir I. (February 1966). "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady*. **10** (8): 707–710.
- [85]M. Keshavarz and Y. Lee, "Ontology matching by using ConceptNet", Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference, 2012.
- [86]"Lucene 3.0.3 API", [Lucene.apache.org](http://lucene.apache.org), 2017. [Online]. Available: http://lucene.apache.org/core/3_0_3/api/contrib-snowball/. [Accessed: 24-jun- 2017].
- [87] R. Rada, H. Mili, E. Bicknell and M. Blettner. (1989). "Development and Application of a Metric on Semantic Nets", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19.
- [88] H. Bulskov, R. Knappe and T. Andreasen. (October 2002). "On Measuring Similarity for Conceptual Querying", Proceedings of the 5th International Conference on Flexible Query Answering Systems; Copenhagen, Denmark.
- [89] Z. Wu and M. Palmer. (June 1994). "Verb semantics and lexical selection", Proceedings of 32nd annual Meeting of the Association for Computational Linguistics; Las Cruces, New Mexico.
- [90] D. Lin. (July 1998). "An information-theoretic definition of similarity", Proceedings of the 15th International Conference on Machine Learning; Madison, Wisconsin, USA.
- [91] G. Varelas, E. Voutsakis, P. Raftopoulou ,E. G. M. Petrakis and E. E. Milios,(2005) .“Semantic similarity methods in WordNet and their application to information retrieval on the web”, Proceedings of the 7th annual ACM international workshop on Web information and data management; Bremen, Germany.
- [92]E. Delele, "Topic-based Amharic Text Summarization", Addis Ababa University, 2011.
- [93]F. Berhanu, "Afaan Oromo Automatic News Text Summarizer Based on Sentence Selection Function", Addis Ababa University, 2013.
- [94]Y. Fisseha, "Development Of Stemming Algorithm For Tigrinya Text", Addis Ababa University, 2011.
- [95]"Google Translate", [Translate.google.com](https://translate.google.com/), 2017. [Online]. Available: <https://translate.google.com/>. [Accessed: 07- Jun- 2017].
- [96]An Afan Oro1no-English, English-Afan Oro1no Dictionary. CAMBRIDGE UNIVERSITY,1913.
- [97]E. Zenawi, Amharic-English, English-Amharic dictionary. [Place of publication not identified]: Simon Wallenberg Press, 2007.

- [98]E. Zecarias, Tigrinya – English and English - Tigrinya Dictionary, 1st ed. Halifax, Nova Scotia, 2007.
- [99]L.Kallipolitis, V.Karpis, I.Karali, World News Finder: How we Cope without the Semantic Web. Artificial Intelligence and Applications (AIA), IASTED/ACTA Press, 2007.
- [100]"Common English Words in English List Available as Download", TextFixer, 2017. [Online]. Available: <https://www.textfixer.com/tutorials/common-english-words.php>. [Accessed: 24- July- 2017].
- [101]"geezorg/data", GitHub, 2017. [Online]. Available: <https://github.com/geezorg/data>. [Accessed: 29- July- 2017].
- [102]"FBC - እንኳን ወደ ፋና ብሮድካስቲንግ ኮርፖሬት ድረገፅ በደህና መጡ::", Fanabc.com, 2017. [Online]. Available: <http://www.fanabc.com/>. [Accessed: 7- Aug- 2017].
- [103]"ድም ወያኔ ትግራይ", 2017. [Online]. Available: <http://www.dmtsiweyane.com/>. [Accessed: 13- Aug - 2017].
- [104]"VOA Amharic", ቪ.ኤ.ኤ, 2017. [Online]. Available: <https://amharic.voanews.com/>. [Accessed: 14- Aug - 2017].
- [105]"VOA Tigrigna", ቪ.ኤ.ኤ, 2017. [Online]. Available: <https://tigrigna.voanews.com/>. [Accessed: 21- Aug- 2017].
- [106]P. University, "About WordNet - WordNet - About WordNet", Wordnet.princeton.edu, 2017. [Online]. Available: <https://wordnet.princeton.edu/>. [Accessed: 24- Aug- 2017].
- [107] G. A. Miller. Wordnet: A lexical database for english. Commun. ACM, 38(11):39–41, 1995.
- [108]"Google Code Archive - Long-term storage for Google Code Project Hosting.", Code.google.com, 2017. [Online]. Available: <https://code.google.com/p/ws4j/>. [Accessed: 25- Aug- 2017].
- [109]M. Porter, "'Lovins revisited", In Tait, J.I. (editor) Charting a New Course: Natural Language Processing and Information Retrieval", Essays in Honour of Karen Spärck Jones, 2017.
- [110]M. Porter, "Stemming algorithms for variousEuropeanlanguages", Snowball.tartarus.org, 2017. [Online]. Available: <http://www.snowball.tartarus.org/texts/stemmersoverview.html>. [Accessed: 02- Sep- 2017].
- [111]R.Schinke, M.Greengrass, A.M.Robertson and P.Willett, "A stemming algorithm for Latin text databases", Journal of Documentation, 1996.

- [112]K.Mohammed, R.Babu and Y.Assabie , “ Afaan Oromo News Text Categorization using Decision Tree Classifier and Support Vector Machine: A Machine Learning Approach”, International Journal of Computer Trends and Technology (IJCTT),2017.
- [113] C.Leacock, and M.Chodorow ,”Combining local context and WordNet similarity for word sense identification”,In Fellbaum, C., ed., WordNet: An electronic lexical database. MIT Press. 265–283, 1998.
- [114]Z.Wu, and M.Palmer, “Verb semantics and lexical selection”, In 32nd Annual Meeting of the Association for Computational Linguistics, 133–138, 1994.
- [115]P.Resnik, “using information content to evaluate semantic similarity in a taxonomy”, In Proceedings of the 14th International Joint Conference on Artificial Intelligence, 448–453,1995.
- [116]D.Lin, “An information-theoretic definition of similarity”, In Proceedings of the International Conference on Machine Learning, 1998.
- [117]J.Jiang and D.Conrath ,”Semantic similarity based on corpus statistics and lexical taxonomy”, In Proceedings on International Conference on Research in Computational Linguistics, 19–33, 1997.
- [118] P.Resnik ,”Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language”, Journal of Artificial Intelligence Research **11**, 95–130 ,1999.
- [119]A. Essayeh and M. Abed, "Towards Ontology Matching Based System Through Terminological, Structural and Semantic Level", Procedia Computer Science, vol. 60, pp. 403-412, 2015.

List of Appendixes

Appendix A. List of stopwords for supported text classifier languages

(I). the corpuses of Amharic stop words

ሁሉ	ያ	እንደተገለጸው	ነበረች	ማድረግ	ብለዋል	ከጋራ	አብራርተዋል
ብቻ	ሆኑ	ጋራ	እንዲሁም	ነገር	ሲል	አመልክተዋል	ወደፊት
አንድ	የተለያዩ	ሌላ	ጥቂት	እያንዳንዳው	አለ	ብለዋል	በሰሞኑ
የሰሞኑ	እዚህ	ተከናውኗል	ማለቱ	ተናግረዋል	ከመካከል	አሳሰበ	አብራርተው
ሁሉም	ይታወሳል	እንደተናገሩት	ነበሩ	ማን	ስለሆነ	ከፊት	ውስጥ
በተለይ	ሆኖም	ግን	እንጂ	ነገሮች	ሲሉ	ይናገራለ	በታች
አንጻር	ተባለ	ሌሎች	ፊት	እያንዳንዱ	አስታወቀ	ብቻ	አስረድተዋል
የታች	እና	ችግር	መካከል	የገለጹት	ከሰሞኑ	አሳሰበዋል	ውጪ
ኋላ	ይህ	እንደአስረዱት	ነበረ	ማንም	አቶ	ወዘተ	በኋላ
በተመለከተ	ሁል	ገሌጿል	እዚህ	ናት	ስለ	ብዛት	እስከ
እስኪደርስ	ተገለጸ	ልዩ	ደግሞ	ከ	አስታውቀዋል	አስፈላጊ	ያለ
የውስጥ	እንደ	ታች	የሚገኙ	ይገልጻል	ከታች	ወይም	በኩል
ሁኔታ	ደግሞ	እንደገና	ነው	ሰሞኑን	ሆኖም	ብዙ	እባክህ
በተመሳሳይ	ሁሉንም	ገልጸዋል	እዚያ	ናቸው	ቢቢሲ	አስገንዘቡ	ያሉ
እንኳ	ተገልጿል	መሆኑ	ዛሬ	ከኋላ	አስታውሰዋል	ወደ	በውስጥ
የጋራ	እንደገለጹት	ትናንት	የሚገኝ	ሲሉ	ከውስጥ	ቦታ	እባክሽ
ሆነ	ደረሰ	ወቅት	ነይ	ሲሆን	መግለጹን	አስገንዝበዋል	ይገባል
የተለያየ	ላይ	ግዜ	እያንዳንዱ	አሁን	ቢሆን	ዋና	በጣም
እስከ	ተጨማሪ	ማለት	ጋር	ከላይ	እስካሁን	በርካታ	እባክዎ
ተቀምጧለች	ችላለች	ነው					

(II). the corpuses of Afaan Oromo stop words

waan	iseen	Fi	Tanaafuu	Akka	Yoom
ofii	isaa	Immoo	Waan	Ituu	Eegana
akka	akka	Moo	Itumallee	Odoo	Silaa
Kun	kan	Illee	otumallee	Silaa	Eega
sun	koo	Akka	Ituullee	Yeroo	Nuti
An	kee	Jechuu	Otuullee	Hanga	Tawullee
kan	Ammo	Jechuun	Ennaa	Erga	Isee
inni	Garuu	Jechaan	Henna	Osoo	Keeti
isheen	yookaan	Osoo	Innaa	Ishee	Otuu
isaan	yookiin	Odoo	Hoggaa	Kan	Utuu
Nu	akkasumas	Ituu	Oggaa	Kun	Otuma
nuyi	Booda	Akkum	Hogguu	Eegasii	Ka
keenya	Erga	Akkuma	Yeroo	Yookinimoo	Yoo
keenya	Eega	Booda	Yommuu	Utuu	Akkasumas
koo	kanaaf	Booddee	Yammuu	Kanaaf	Ofii
kee	kanaafi	Dura	Yemmuu	Tahullee	Malee
sun	kanaafuu	Kanaafi	Yommii	Akkam	Erga
ani	tanaaf	Saniif	Simmoo	Otoo	Erga
Ini	tanaafi	Tanaaf	Oo	Iseen	Waggaa
Isaan	tanaafuu	Tanaafi	Woo	Keetii	Oggaa

(III). the corpuses of Tigrigna stop words

ኣብ	አለዋ	ብዘይካ	እዙይ	እንታይ	ከምቲ	ከማኩም	ከምዘለኪ
ናይ	አለዎ	ብኡ	እምበር	እንከለና	ከምናቶም	ከማክን	ከምዘለወን
ካብ	አመት	ብሰንኪ	እንትኸውን	እቲአ	ከምናታ	ከምኣም	ከለና
እቲ	አመታዊ	ብምባል	እንተኮነውን	እቲአም	ከምዘይብልና	ከማካ	ክኸውን
ምስ	አይካአለን	ድህሪ	እንተኮነግን	እቲአቶም	ከምዘይብሎም	ከማኪ	ልእሊ
ከም	አይኮነን	ድህሪ ህዚ	እንተኮነግና	እቲአተን	ከምዘይብለይ	ከምኣቶም	ምስ
ድማ	ብስቡቕ	ድህሪት	እንተይኮነስ	እቲአን	ከምዘይብልኪ	ከምኣተን	ማለታ
ናብ	በዚ	ድህሪትን	እስኪ	እቲይ	ከምዘይብልካ	ክስቶ	ማለቱ
እዩ	ብተወሳኪ	ድማ	እምበር	እዙይ	ከምዘይብልክን	ክስታይ	ማለተይ
ከአ	ብተወሳኺ	ደአ	እሎም	እዚአ	ከምዘይብልኩም	ከምዚኣም	ማለትኪ
ግን	ብአምሆይ	ግና	እሲ	እዞም	ከምዘይብላ	ካብዚኣም	ማለትኩም
እዚ	ብዘይ	ግዳ	እሎም	እዚአተን	ከምዘይብሉ	ከምዚአን	ማለተን
ሓደ	ብዘይካ	ገገለ	እሉ	እዚኣቶም	ከምዘይብለን	ከምዝኸኑ	ማለትክን
ነቲ	ቦቲ	ገለ	እልና	እዚ	ከአ	ከምኣን	ማለትና
አብቲ	ቦታ	ገለገለ	እለና	እዚአ	ካብዚ	ከምዙይ	ማለቶም
አነ	ቦቶም	ህጂ	እልክን	እዚኣም	ከምዛ	ከምዚአ	ማለትካ
አበይ	ቦተን	ህዚ	እልኩም	እዚአን	ከምቲ	ከምዚኣቶም	መአዝ
አየን	ብቶም	ሃልሃሊፉ	እለ	እንታዋይ	ከምተን	ከምኡ	መን
አላ	ብተን	ህድህድ	እላተን	እንታወይቲ	ከምቶም	ካብታ	ምሳይ
አብዚ	ቦቲአ	ህዚ	እዙይ	እንታዎት	ከምታ	ኮነ	ምሳኪ
አባይ	ቦቲአን	ህዚ	እንተሎ	እከለ	ከምቲኣም	ካብቶም	ምሳካ
አባኪ	ቦቲአተን	ሃደሃደ	እንተላ	እገለ	ከምቲአን	ኳ	ምሳኩም
አባካ	ቦአና	እንትኸውን	እንተለኩ	እንትኸኑ	ክሳብ	ክንድቲ	ምሳና
አባካን	ብአካ	እንትኸውን	እንተለኪ	እዚዩ	ክስይ	ከምኡውን	ምስኣም
አባክሙ	ብአኪ	እቲ	እንተለኩም	እንታይነት	ኩሎም	ከምዝኸነ	ምስኣቶም
አብአን	ብአይ	እታ	እንተለው	እንተድአ	ኩልና	ኸነ	ምስአን
አብኣም	ብአኩም	እዩ	እንተለክን	እንኮ	ኩልክን	ኮይኑ	ምሳክን
አባና	ብአክን	እናተ	እንተለና	እንትኸና	ኩላትና	ኮይና	ምሳካትክን
አብኣቶም	ብእኣም	እዮም	እንተለካትኩም	እስካብ	ክላቶም	ኮይኖም	ምስመን
አብአተን	ብአተን	እቶም	እንትባሃል	እልካ	ኩሉኩም	ኮይነ	ምስአ

አብዙይ	ብአን	እዋ	እንተኮይኑ	እንተዘይኮነ	ኩለካትክን	ከዚ	ምስኡ
አብዚ	ብአካትኩም	እዙይ	እንተኮይነ	ካብ	ኩላተን	ኸዚ	ምስቶም
አነ	ብአካትክን	እውን	እንተዘይኮነ	ካሊእ	ከመይ	ከምዘለዋ	ምስታ
አዝዩ	ብኩሎም	እዚ	እንተኮይና	ክንደይ	ካልኦት	ከምዘለዉ	ምስቲ
አንተዎ	ብኩልና	እያ	እንተኮይኑም	ክንዲ	ከምዚ	ከምዘለና	ምስተን
አበይ	ብኩላኩም	እያተን	እንተኮይኖም	ከም	ከምኡ	ከምዘለኩም	ምእንቲዚ
አብኡ	ብኩልክን	እያቶም	እንተኮይንካ	ከከም	ከማይ	ከምዘለክን	ምስዚ
አብዛ	ብብሃደ	እየ	እንተኮይንኪ	ክብል	ከምኦ	ከምዘላ	ምስ
አብዚ	ብአክን	ሀዚ	እንተኮይንን	ኩሉ	ከማና	ከምዘለኩ	ምስምስ
ንኩለ	ነጀው	ናይመን	ንኦኦ	ነዞም	ሰለዚ	ሰለዘየላ	ዘሎ
ንኩሉ	ናታተን	ነተን	ንኦና	ንዞም	ሰለዚዝኸነ	ሰለዘለና	ዘላ
ንኸሉ	ናታቶም	ናፍቲ	ንእኦም	ንዘን	የለን	ሰለዘሎ	ዘለዋ
ንኩሎም	ናትኪ	ናፍታ	ንኦተን	ንዘለዎም	የላን	ሰቡቕ	ዘለዉ
ንኩለን	ናትካ	ናፍቶም	ንኦክን	ንዘለወን	የለኩም	ጥራይ	ዘለካ
ንኩልና	ናትኩም	ናፍተን	ንኦኩም	ንዘለና	የለካን	ጥራህ	ዘለኩ
ንኩላትና	ናታትኩም	ናባይ	ንኦኦቶም	ናይዚ	ይኩን	ታህቲ	ዘለኩም
ናብቲ	ናትክን	ናባኪ	ንኦካትክን	ናይዛ	የለናን	ውን	ዘለኪ
ናይቲ	ንስካ	ናባና	ንኦኦን	ናይዞም	የለውን	ወትሩ	ዘለና
ነቲ	ንሱ	ናባካ	ናብዚኦ	ናይዘን	የለዋን	ወይወን	ዘለክን
ናይ	ንሳ	ናብኦ	ናብዚኦም	ናብቶም	ይኩንደኦ	ወይ	ዘይብሉ
ነቶም	ንሶም	ናብኡ	ናብዚኦን	ንቶም	ይኩንደኦምበር	ዉን	ዘይብላ
ናብ	ንሀና	ናብኦም	ነዛ	ንተን	ይኩንምበር	ወይከ	ዘይብለይ
ነቱይ	ንሳቶም	ናብኦን	ነዚ	ናታትክን	የልቦን	ወላእክዋ	ዘይብልካ
ነታ	ንሳተን	ናባክን	ነዙይ	ነናይ	ይኦይ	ዋላ	ዘይብልኪ
ነዚ	ንስኪ	ናባኩም	ነዚኦ	ንብምሉኦም	ቅድሚት	ወላዉን	ዘይብሎም
ናቱ	ንስክን	ናብኦቶም	ነዚኦም	ነናተን	ቅድሚ	ውሁዳት	ዘይብለን
ናተይ	ንሰን	ናብኦተን	ነዚኦን	ንላእለዎት	ሰለዘየለዉ	ውሁድ	ዘይብልና
ናተን	ንስኩም	ንኦይንኦኪ	ነዚኦቶም	ላእለዎት	ሰለዘሎ	ዘለውኻ	ዘይብልኩም
ናተና	ንምንታይ	ንኦካ	ነዚኦተን	ሰለዝኮነ	ሰለዘላ	ዝኸውን	ዘይብልክን
ናቶም	ንመን	ንኡኡ	ነዘን	ሰለዙይ	ሰለዘየለ	ዝኸነት	ዘይኮነስ

Declaration

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

Tsegay Mullu Kassa

This thesis has been submitted for examination with my approval as an advisor.



Yaregal Assabie, PhD
Addis Ababa, Ethiopia

Jimma, Ethiopia November 2017