



JIMMA UNIVERSITY
SCHOOL OF GRAGUATE STUDY
JIMMA INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

GENERAL PURPOSE LANGUAGE IDENTIFICATION FOR
ETHIOPIA SEMITIC LANGUAGE USING HYBRID APPROACH

KIDST ERGETIE ANDARGIE

A THESIS SUBMITTED TO
THE SCHOOL OF COMPUTING OF JIMMA UNIVERSITY IN PARTIAL
FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF
SCINECE IN INFORMATION TECHNOLOGY

Jimma, Ethiopia
November 2017

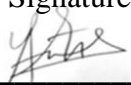
JIMMA UNIVERSITY
SCHOOL OF GRAGUATE STUDY
JIMMA INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

GENERAL PURPOSE LANGUAGE IDENTIFICATION FOR
ETHIOPIA SEMITIC LANGUAGE USING HYBRID APPROACH

KIDST ERGETIE ANDARGIE
Advisor: DR.YAREGAL ASSABIE

This is to certify that the thesis prepared by *Kidst Ergetie Andargie*, titled: *General Purpose Language Identification for Ethiopia Semitic Language using Hybrid Approach* 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>
Examiner: _____		
Examiner: _____		

Abstract

Due to many sophisticated and advanced technologies like the Internet, the world has become a single village. It is possible to get a vast amount of digitized information that are generated, propagated, exchanged, stored and accessed through the internet and other media like mobile network each day across the world. The accumulation of digital data is making information acquisition increasingly difficult, with natural language becoming critically an obstacle. The step towards tackling this obstacle is Natural Language Processing and language identification is the first step among many steps that are used for information acquisition and other advanced NLP applications. It is a technique of labeling each word in a text or sentence with its corresponding language category. In past decades a number of research works have been done in the area of language identification. However, there are issues which are not solved until: multilingual language identification, discriminating the language category of very closely related language documents and labelling the language category for very short texts like words or phrases. In addition to this, as far as the researcher's knowledge is concerned, there is no language identifier developed for Ethiopian Semitic language though there are many language identifier developed using different approaches for many European languages and resourced languages.

In this investigation, we propose a hybrid approach; character ngram and word ngram combined with rule based approach. Which can able to solve these mentioned unsolved issues of language identification on top of Ethiopian Semitic languages (i.e. Amharic, Geeze, Guragigna and Tigrigna). The proposed general purpose language identifier approach has a capability of identify the language of a text at any level (i.e. Word, phrase, sentence and document) in both monolingual as well as multilingual setting. The reason behind this capability of proposed approach is due to the features of word level language identification, in which every words needs to classify with regard to its language category at a time. Text is first pass through preprocessing steps. Then pass through rule based approach word which can handle through rule. Afterwards word ngram of prewise word language is conducts, if word not exist, Character ngram (infinite ngram) with location is calculated; afterwards the ngram probability is calculated and ngram probability of word is calculated, which is used to assign a language label for that word. Finally sentence and document reformation is done for all texts.

The system was developed using Java programming and the performance of the system has been evaluated using 10-fold cross-validation technique. For training and testing purpose 27 Mb data from different sources (news, bible and books) were used. Beside this, the effectiveness and performance of the proposed language identifier is evaluated using precision, recall and F-measure evolution metrics. Different experiments are conducted for hybrid of character ngram, rule based and word ngram based approaches using monolingual texts. The hybrid of fixed size character ngram with location, word ngram and rule based approach shows an average F-measure of 70.39%, 76.95 % 4, 73.69 % and 78.98% for Amharic, Geez, Guragigna and Tigrigna respectively. The hybrid of infinite ngram with location, word ngram and rule based approach shows an average F-measure of 83.57%, 84.53%, 86.67% and 87.44% for Amharic, Geez, Guragigna and Tigrigna respectively. Whereas, the hybrid model (adding sentence) improve the accuracy to 99.85%, 99.74%, 100% and 99.93% for Amharic, Geez, Guragigna and Tigrigna respectively. Adding sentence and document reformation improves the performance in to 100% for word, phrase, and sentence and document level in a monolingual setting. As well, for multilingual setting also attains an average F-measure of 100% for both sentence level and document level test, but for phrase level achieves an average F-measure of 82.64%, 86.38%, 87.19% and 86.81% For Amharic, Geeze, Guragigna and Tigrigna respectively. Hence, it is found that adding sentence level and document level reformation in to the hybrid of infinity ngram with location feature set is a best combination of proposed general purpose language identifier.

Key words: language identification, multilingual, monolingual, Naïve Bayes, ngram, closely related language, ngram location, word level, infinity ngram, fixed length character ngram

Dedicated to:

1. My Mother Kassanesh Ayalew

Mom!! You were dedicated to change my life! You have walked all the way you can to change and to educate me though you did not see your effort! May your soul rest on heaven!

2. My Father Ergetie Andargie

Dad!! You made me strong, motivated and passionate person. I am here because of your inspiration during my childhood though you are not lucky to see your passion, courage and motive in your child. May your soul rest on heaven!

3. My husband Tsegay Mullu

I wish I could call you everything for me! My brother, my father my mother... you have helped me in my entire career.

Acknowledgment

Let all the praise and thanks be to the supernatural power and creature of the entire universe, almighty God for helping me to realize this work. Then, I was very lucky enough to have had the support of many people and without these people the completion of this thesis work would have been very thorny. Firstly I would like to thank my advisor **Dr. Yaregal Assebie**. I thank you **Dr.** for your encouragement, guidance, understanding and motivation throughout this thesis work. You have showed me how researches are produced and how to tackle a problem by investing your invaluable time. I really thank you! Throughout this thesis work, you have been consistently sharing me constructive ideas and comments about this work in a friendship mode which I love most. Really you are a model advisor. I would like to thank my co-advisor Mr. Shimelis Shiferaw. Special thank goes to my husband **Tsegay Mullu** who has been always with me during the entire of this work; without him, it would be impossible to finish this work. It is also my pleasure to express my gratitude to my sisters and many of my friends who have helped me in this thesis work.

Finally, I would like to thank everyone who has contributed negative and positive impacts to the successful realization of this thesis work, as well as expressing my apology that I could not mention individually one by one.

TABLE OF CONTENT

Contents	Page
Abstract.....	i
List of Figures.....	vii
List of Tables.....	vii
Acronyms and Abbreviations.....	viii
Introduction.....	1
1.1 Background.....	1
1.2 Statement of the Problem.....	4
1.3 Objectives.....	5
1.3.1 General Objective.....	5
1.3.2 Specific Objective.....	5
1.4 Methods.....	5
1.4.1 Literature Review.....	5
1.4.2 Data Collection.....	5
1.4.3 Design the Framework.....	6
1.4.4 Prototype Development.....	6
1.4.5 Evaluation.....	6
1.5 Scope.....	6
1.6 Application of Results.....	7
1.7 Thesis Outline.....	7
Literature Review.....	8
2.1 Introduction.....	8
2.2 Language Identification Feature Types.....	9
2.3 Using Words as Feature Type.....	10
2.3.1 Frequent Words Method.....	10
2.3.2 Lexicon Method.....	11
2.3.3 Short Words Method.....	11
2.4 Using N-grams as Feature Type.....	11
2.5 Language Identification Approaches.....	12
2.6 Computational Approach.....	12
2.6.1 Bayesian Classification.....	14
2.6.2 N-gram Method.....	15
2.6.3 Markov Model.....	17
2.6.4 Support Vector Machine.....	19
2.7 Non Computational Methods.....	20
2.8 Hybrid Approach.....	21
2.9 Empirical Evaluation.....	21
2.10 Multilingual Documents.....	24
2.11 Closely-related Languages.....	26
2.12 Ethiopian Semitic language.....	27
2.13 Summary.....	30

Related Work	31
3.1 Introduction.....	31
3.2 Language Identification for Monolingual.....	32
3.3 Language Identification for Multilingual.....	35
3.4 Language Identification for Closed-related Language	37
3.5 Summary.....	39
Design of Proposed Language Identifier	42
4.1 Introduction.....	42
4.2 Design Goals.....	42
4.3 Architecture of General Purpose Language Identifier	43
4.3.1 Document Loading	44
4.3.2 Sentence Segmentation.....	45
4.3.3 Tokenization	46
4.3.4 Indexing.....	48
4.3.5 Normalization	48
4.3.6 Character Ngram Extraction.....	50
4.3.6.1 Fixed Length Character ngram.....	52
4.3.6.2 Infinity ngram.....	54
4.3.7 Word ngram	60
4.3.8 Rules	62
4.3.9 Classification.....	65
4.3.9.1 Additive Smoothing.....	67
4.3.10 Sentence Level Reformation	67
4.3.11 Document Level Reformation.....	70
4.4 Prototype.....	71
4.5 Summary.....	73
Experiment.....	74
5.1 Introduction.....	74
5.2 Data collection	74
5.2.1 Data Sources	75
5.2.2 Bible.....	75
5.2.3 Books.....	76
5.2.4 News.....	78
5.2.5 Data Cleaning.....	79
5.3 Cross-Validation	79
5.4 Implementation	82
5.5 Evaluation	83
5.5.1 Evaluation Metrics.....	83
5.5.2 Test Result	83
5.6 Discussion.....	89
Conclusion and Recommendation	93
6.1 Conclusion	93
6.2 Contribution	95
6.3 Recommendation	96
References.....	97

List of Figures

Figure 2.1: A general architecture of a language identifier based on computational approach....	13
Figure 2.2: The graph resulting from the example training set.....	16
Figure 2.3: The graph resulting from the evaluation text	16
Figure 2.4: Example of the rank-order statistics classifier	17
Figure 4.1: General Architecture of Proposed Language Identifier	43
Figure 4.2: Text area of prototype after text document is loading.....	45
Figure 4.3: Character trigram representation of word “አንዳይከሰት”	52
Figure 4.4: infinity ngram character representation of word “አንዳይከሰት”	55
Figure 4.5. The Proposed General Purpose Language Identifier Prototype User Interface	71

List of Tables

Table 2.1: Most frequent words of European languages	10
Table 3.2: Comparison of the results using article and training model lengths Cavnar and Trenkle work	32
Table 4.1: Normalized Characters	49
Table 4.2: Guragigna specific characters.....	63
Table 4.3: Tigrigna and Guragigna unique characters.....	63
Table 5.1: Books which are written in Amharic language.....	77
Table 5.2: Book which are written in Tigrigna language	77
Table 5.3: Book which are written in Geez language	78
Table 5.4: Book which are written in Guragigna language	78
Table 5.5: total Corpus size after data cleaning.....	79
Table 5.6: Corpus size for training the models (90%).....	81
Table 5.7: Corpus size for testing the models (10%).....	81
Table 5.8: Corpus size for word based approach.....	81
Table 5.9: Statistics of test data corpus.....	82
Table 5.10 : Experimental Results for experiment 1	84
Table 5.11 : Experimental Results for experiment 2	85
Table 5.12 : Experimental Results of infinite length ngram	87
Table 5.13 : Experimental Results of infinite length ngram with location feature	88
Table 5.14 : Experimental Results of sentence reformation feature	89
Table 5.15 : Experimental Results of monolingual texts	88
Table 5.16 : Experimental Results of multilingual texts.....	89

Acronyms and Abbreviations

KDT	Knowledge Discovery in Text
LangID	Language identification
LIGA	Language Identification Graph
MLIR	Multilingual Information Retrieval
NLP	Natural Language Processing
OOP	Object Oriented Programming
SVM	Support Vector Machine

Chapter One

Introduction

1.1 Background

Textual data are getting more and more available on the global network. This data's written in number of different language. In order to use the content of these textual data, one should know the language in which it is written or it has to be translated to the local language or mother tongue of an individual, which needs a language translator. Added to this is the collection of multilingual documents available in digital form which is quite natural in a multilingual country like Ethiopia.

Different languages have different grammatical structures; the language processing tools are language dependent. Hence, there is need for automated tools and techniques which can identify the language of the written text and then select the required tools for further processing of the text based on the language of the written text. The solution to this problem is the Language Identification. Language identification (LangID) is the process of identifying a language in which a text document is written. The problem of LangID is one that is intuitively familiar, since one of the characteristics of being human is the ability to communicate complex and sophisticated thoughts and ideas, and this is only possible through the use of a common language. People are generally quickly able to recognize languages that they are familiar with.

Research into LangID aims to mimic this human ability to recognize specific languages. Over the years, a number of computational approaches have been developed that, through the use of specially-designed algorithms and data structures, are able to infer the language being used without the need for human intervention [1]

An average person may be able to identify a handful of languages and a trained linguist or translator may be familiar with dozens, but most of us will have experienced at some point an encounter with a language that is alien to us. However, LangID research aims to develop systems that are able to recognize any human language, a set which numbers in the thousands [1].

The ability to accurately detect the language that a document is written in is an enabling technology that increases accessibility of data and has a wide variety of applications. In natural language processing (NLP), most techniques presuppose that the language of input text is known, and many techniques further assume that all documents are in the same language. In order to apply NLP techniques to real world data, LangID is typically the first step in order to ensure that only documents in relevant languages are subjected to further processing. Similarly, in information storage and retrieval, it is common to index documents in a multilingual collection by the language that they are written in, and LangID is necessary for document collections where the languages of documents is not known a-priori, such as in data crawled from the World Wide Web [9].

Another application of LangID that predates computational methods is the detection of the language of a document for routing to a suitable translator; this application has become even more prominent due to the advent of machine translation methods. In order for machine translation to be applied to translate a document to a target language, it is generally necessary to know the language being translated from, and this is the task of LangID. LangID also plays a part in helping to bridge an increasing “digital divide” by providing support for the documentation and use of lower-density languages [9].

Collection of Text documents on the web which gives for language identification as input may be written in only one language or in multiple languages. If you compare the two situations Processing monolingual documents is fairly simple compared to multilingual documents. Because knowing one language and knowing several languages are quiet difference [3].

Basic Multiple languages identification (multilingual LangID) Challenges are, According to [3]:

- **Segmentation of documents:** - the segmentation of documents which identifies the regions of a document in different languages is a problem to processing multilingual documents. Once the region is identified, the language of the content in that region can be identified and used for further processing [3].
- **Common words:** – In a multilingual country like Ethiopia, vocabulary of a language gets influenced by various other languages and in due course of time those words become part

and parcel of the language. Further, in case of similar languages, certain words are used commonly in all languages [3].

- **Closely related languages:** - Similar languages or dialects of languages form the closely related languages and share a great deal of lexical and grammatical features.

Not only those challenges also; In multilingual LangID It is mandatory first to check whether the text document is written with monolingual or multilingual language before language identification applies. For multilingual identifier knowing *language switching* is big challenge. This specifies how frequently or where a shift from one language to another can occur in a document [3].

Research to date on LangID has generally focused on *monolingual* documents [2]. In monolingual LangID, the task is to assign each document to a unique language. Increased availability of large quantities of textual data from a diverse variety of sources has led to a demand for methods to identify language in settings which diverge greatly from those that have been examined in the literature. So, now a day's multilingual documents identification is a hot research area. As far as my knowledge there is limited number of works done on multilingual documents. However, there is no research conducted for Ethiopian language except the one that works on Cushitic language for monolingual text [9]. Even if a different research works are conducted in the area of language identification in a past decades. However, there are issues which are not solved until now: multilingual language identification, identify the language of texts written in very closely related languages, and languages for very short texts (i.e. words and phrases). Hence, in this research, we propose an approach which can able to solve these unsolved issues of language identification in once. Beside this, the proposed approach also shows a performance comparison between fixed character ngram with infinity ngram combine with word ngram as well as rule based approach. In all combinations, the ngram relative frequency and location features sets used in order to navigate their effect in the proposed approach.

In general, in this investigation the proposed general purpose language identifier aims on top of Ethiopian Semitic language (i.e. Amharic , Geeze , Guragigna and Tigrigna) to identify the language category of input text at any level (i.e. word , phrase , sentence and document) at both monolingual as well as multilingual setting.

1.2 Statement of the Problem

Written Documents in Ethiopian Semitic languages are getting larger and larger in volume both online and offline since the languages are used in regional or zonal education curriculum and as working languages in regional or zonal offices. To increase accessibility of data on the internet for those languages (Amharic, Tigrigna, Guragigna and Ge'ez) need language identifier.

In addition to this, in order to apply many multilingual NLP application like: text mining, identification of the language www pages, information retrieval systems, and content based and language specific web crawlers, search engines, and online language translation application etc identify text documents of the language is required. However; there is no work done before for identification of Ethiopian Semitic languages. So, developing a language identifier for Ethiopian Semitic languages is one aim of this research.

Beside this, in Computational Linguistics (CL), language identification of text is generally considered as solved problem but there are issues which are not solved until now. Identification of a language for a multilingual texts separately is one of the unsolved issues in language identification, since previous researchers used character window size based approach and this approach is ambiguous to discriminate the switching point of a language. A single word character portions can be assigned to one or more language categories because the character window size is done randomly. Another unsolved issue in language identification is identifying language category for very closely related languages, since these languages have words in common and they have same script. In addition to this, they also usually share sequence of characters in common. On the other hand, in state of the art of language identification identifying a language category for very short texts (i.e. word, phrase) is unsolved hot research issue. Since the features extracted from such very short texts are not rich in order to make decision for language category of a given text.

Hence, in this investigation we aim to solve these unsolved language identification issues by designing a framework of general purpose language identifier on top of Ethiopian Semitic languages (i.e. Amharic, Geeze, Guragigna and Tigrigna), which able to identify the texts at any input level (i.e. word, phrase, sentence and document) for both monolingual and multilingual setting.

1.3 Objectives

1.3.1 General Objective

The general objective of this research work is to investigate general purpose Language Identification model for Ethiopian Semitic Languages, particularly for Amharic, Geeze, Guragigna and Tigrigna.

1.3.2 Specific Objective

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

- Study transcriptions of Ethiopian Semantic languages
- Assess different techniques and approaches employed so far in Language Identification.
- Design and develop a model for LangID system for Ethiopian Semitic language
- Organizing training and test corpus data
- Develop a prototype for LangID system
- Conduct experiment to evaluate the prototype performance
- Give conclusion and recommendation based on experimental result

1.4 Methods

The following methods are applied in order to achieve the above specified objectives.

1.4.1 Literature Review

Different information resources and related works are reviewed. This may consists of conference and journal articles, white papers and LangID systems developed for other languages. In addition, there may be a discussion with Linguistic experts regarding the linguistic nature of the languages like the grammatical structure and the properties of Semitic languages.

1.4.2 Data Collection

The approaches' requires a corpus data to develop the LangID. The data to be used for text corpus are obtained from various sources of Ethiopian News. These texts will collect from different sources both in softcopy and in hardcopy. Such sources include websites of Ethiopia

Broadcast Corporation (EBC), Fana Broadcast Corporation (FBC), books and others written text is used.

1.4.3 Design the Framework

To conduct this research we are use hybrid approach by tacking best features of word based Ngram (WBN), character based Ngram (CBN) and Rule based approach.

1.4.4 Prototype Development

In order to develop LangID prototype java programming language is used, since java has good computing performance than other programing language. A lot of Java efficiency comes from optimizations to virtual machine execution. It is easy to build and has pure object oriented features. In order to evaluate and show the usability of the proposed work, General purpose language identifier is developed and its performance is evaluated.

1.4.5 Evaluation

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

1.5 Scope

LangID applies to any modality of language, including speech and handwritten text, and is relevant for all means of information storage that involve language, digital or otherwise. However, in this thesis, we limit the scope of our investigation to LangID of documents.

The models for the research is trained and tested on four languages Amharic, Tigrigna, Guragigna and Ge'ez not include others sematic language because of resource and time limitation. Beside this, for all these supported language of the proposed word level language identifier it can identifies

- ✓ **Monolingual document LangID identification:** a document which is written in one language
- ✓ **Multilingual document LangID identification:** a document which is written in more than one language.
- ✓ **Short texts:** which includes word and phrase

LangID identifies a text in three levels: at word level, at sentence level and at document level. It also detects where the language switch from one language to other language.

1.6 Application of Results

It is mostly used as an important preprocessing step for multilingual NLP application. But LangID plays a major role in several NLP applications. Such as Machine translation, Part of Speech tagging, linguistic corpus creation, supporting low-density Languages, accessibility of social media/user-generated content, search engines information extraction, e-mail routing and filtering engines, text mining applications, identification of the language or encoding of WWW pages, information retrieval systems, content based and language specific web crawlers and search engines and spell checker applications.

1.7 Thesis Outline

This thesis contains seven chapters. Chapter 2 gives a comprehensive literature review of LangID. Chapter 3 discuss about the related work done in the area of LangID. Chapter 4 discusses overview of data collection. Chapter 5 is the broad and the crucial part of the research, which discuss about architecture, design, and implementation of the proposed system. Chapter 6 discusses multilingual LangID evaluation techniques, evaluation result and discussion, the performance gap between different approaches. The last chapter discusses the conclusion, recommendation and future work forwarded.

Chapter Two

Literature Review

Within this chapter we provide a brief description about concepts that mainly characterize our thesis: Language identification and issues related with language identification.

2.1 Introduction

The wide use of numerical data and textual information facilitates the sharing of information between people, where the important size of the shared information leads to increase the size of such textual information and databases in different fields. Hence, the access to the information becomes difficult or expensive, and in this respect, many research works were performed to extract the information automatically from databases. That task needs a natural language processing (NLP) or computational linguistics based processing of the written texts, which requires knowing the language in advance, to select the best features and the appropriate language processing procedures. Then, language identification is one of an important step in the information extraction process. That is the reason prompted many researchers to deal with the field of language identification during the last years [2] [17].

Language identification (LangID) is one of the NLP applications [18] it can be also seems as a specific instance of the more general problem of an item classification through its attributes. Text documents are classified by language identification method based on language categories.

LangID is the problem of determining which natural language given content is in or it is a task of detecting languages of given content [18]. LangID is one of the basic steps needed for preprocessing text documents in order to do further textual analysis. For example to identify the language of the following two sentence programmatically.

What is the language of the following sentences written in?

ዓለማችን ላይ ስልሳ ሺህ የዛፍ ዝርያዎች እንዳሉ አንድ ጥናት አስታወቀ፡፡

አብ ዓለምና ስልሳ ሺህ ናይ ኣሞ ዘርእታት ከምዘለው ሐደ ፅንዓት አፍሊጡ፡፡

The process of assigning a sentence or document or short text to classes or language categories that are represented by a finite set of labels is not an easy task. However, it is possible to be carried out through computational methods, since natural languages are extremely nonrandom, and they have regularities in the use of characters or character sequences. So, according to this the first sentence of the above example to be classified as the first textual document written in Amharic and the second one classified as Tigrigna. The alphabet of each language is either unique or highly characteristic of this language. Information on the stability and consistency of the frequency of letters and letter sequences are not new [19]. It is statistically proven that for each language, the number of occurrences of the sequence of two, three, four or five letters are stable and different from language to language.

Being able to identify the language of a given text proves useful in a large number of applications. It can be used in Cross-language Information Retrieval (CLIR) - also known as Multilingual Information Retrieval (MLIR), text mining and knowledge discovery in text (KDT) [20] etc. In addition, in Text-to-speech applications capable of reading multiple languages need to first identify the language correctly. It might prove useful to be able to use automatic translation tools in those cases where the source language is not known. Finally, OCR digitalization of written text can make use of language identification, both to classify the output document and to improve the OCR process itself (as language specific knowledge might be used to improve the accuracy of the conversion). In these fields, the text processing step such as indexing, tokenization, part of speech tagging (POS), stemming and lemmatization are highly dependent on the language.

2.2 Language Identification Feature Types

From a theoretical perspective, the document representation consists of a distribution over the entire space of possible character sequences, whether this space is the space of words or the space of fixed-length sub-sequences. In practice, such a space is either exponentially large (in the case of character n -grams), or infinite (in the case of words), which presents computational challenges. The practical solution to this is to select a subset of sequences which we will consider as being “relevant” to discriminating between languages, a process known as feature selection.

Feature selection provides benefits beyond complexity reduction, which has the advantage of reducing computational resource. There are different feature types for the purpose of language identification task. Some of the most prominent types are discussed in the following sub section:

2.3 Using Words as Feature Type

In order to detect a language a fragment of word can be used as a feature types or can determine the characteristics of a particular language [21]. Word segmentation is usually done by simply tokenizing on whitespace [22], which limits the applicability of such methods to languages where words are whitespace delimited. The bag of words feature type can be implemented through frequency words method, lexicon (unique word) method and short word method.

2.3.1 Frequent Words Method

One of the direct ways for generating language models is to use words from all languages in the training corpus. Due to the Zipf's Law, words with the highest frequency should be used. Such features are used in the frequent words method, where a language model is generated using a specific amount of the words, having the highest frequency of all words occurring in a text or text corpus. The words are sorted in descending order of their frequencies.

For example, Table 2.1 shows the most frequent words generated from the datasets of news collected for the Leipzig Corpora Collection [23]. It is quite obvious that many of these words, such as “de”, “la”, “a”, are shared between more than one language making choosing between them more difficult.

French	German	Spanish	Galician	Dutch	Portuguese	Catalan	Italian
de	der	de	de	de	de	de	di
la	die	la	a	van	a	la	e
le	und	que	e	een	que	que	il
à	in	el	que	en	o	i	la
et	den	en	o	het	e	a	che
les	von	y	do	in	do	el	in
des	mit	a	da	is	da	l	a
en	auf	los	en	op	em	en	per
a	das	del	un	te	para	per	un
l	zu	se	unha	met	os	del	del

Table 2.1: Most frequent words of European languages

2.3.2 Lexicon Method

The Other direct ways for generating language models is to use unique words from the language in the training corpus. For each language all unique words which are found in the corpus should be used in the lexicon method. In this thesis we use this method by mixing with character ngram.

2.3.3 Short Words Method

The short word-based approach is similar to the frequent words method, but it only uses words up to a specific length. Common limits are 4 and 5 letters. Words with this length are mostly determiners, conjunctions and prepositions that are often language specific.

2.4 Using N-grams as Feature Type

Another successful approach for generating language models is the N-gram approach. Cavnar and Trenkle [24] used it for text categorization and found out that it also performed well on the task of language identification. In this approach, a language model is generated from a corpus of documents using N-grams instead of complete words, which are used in the first two approaches.

An **N-gram** is a contiguous N-character slice of a string or a substring of a word and respectively words depending on the size of N [24]. The beginning and the end of a word are often marked with an underscore or a space before N-grams are created. This helps to discover start and end N-grams at the beginning and ending of a word and to make the distinction between them and inner-word N-grams. For instance, the word data, surrounded with the underscores, results in the following:

N-grams:

Unigrams: _, d, a, t

Bigrams: _d, da, at, ta, a_

Trigrams: _da, dat, ata, ta_

Quad grams: _dat, data, ata_

5-grams: _data, data_

6-grams: _data_

To detect the language of a document, at first its N-gram language model is created. Commonly, preprocessing is employed, i.e. punctuation marks are deleted. Moreover, they are tokenized and

surrounded with spaces or underscores. From these tokens, N-grams are generated and their occurrences are counted. The list of N-grams is sorted in descending order of their frequencies and the most frequent ones produce the N-gram language model of the document.

The main advantage of the N-gram-based approach is in splitting all strings and words in smaller parts than words. That makes errors, coming from incorrect user input or Optical Character Recognition (OCR) failures; remain only in some of the N-grams, leaving other N-grams of the same word unaffected, which improve correctness of comparing language models. However, N-grams of small length are not very distinctive and some of them are present in language models of many languages, particularly for those very closely related languages.

To handle this problem, in our thesis the fusions of both Word based and character Ngram feature types are used for general purpose language identification. In addition to this, using character Ngram to determine a language at word level have its own problem, since features extract to represent a given word is small, so in our study a novel approach which able to extract large number of character ngram features for a particular word called Infinity Ngram approach (use all Ngram features of a word in one) is used. We will discuss about this novel approach under design and implementation chapter.

2.5 Language Identification Approaches

In order to classify the textual documents based on language categories, there are three basic by language identification approaches: Computational approach, Non computational approach and Hybrid Approach.

2.6 Computational Approach

Computational approaches are relying on statistical techniques rather than linguistic knowledge to solve related problems. It requires large set of training data for each to be identified language. Computational approaches are subdivided into two phases: I) Training phase and II) Classification phase.

(I). Training phase, for each language the feature extraction either a word or Ngram as feature type are used from the given training corpus to generate a profile (model) for each of given languages is done. A corpus is a large collection of electronically stored written texts. Since the languages are given, this training phase can be classified as supervised learning in the matter of machine learning.

(II). Classification phase, the similarity measure between the training profile and the testing profile is found out and the most similar language is known as the language of the document. In summary, in the classification phase, the language of the given document is classified as follows:

- a. A model is generated for the given document.
- b. Subsequently, the similarity between the document's model and each language models is computed.
- c. The language model, which is most likely, is chosen as the language used in the document.

Finally, the LangID problem is a classification problem with the languages as the classes and the unknown document's language as the query. The questions are which feature types and similarity measure is most suitable?

The general architecture of the language identifier is given by Padró and Padró (2004) in Figure 2.1 [25].

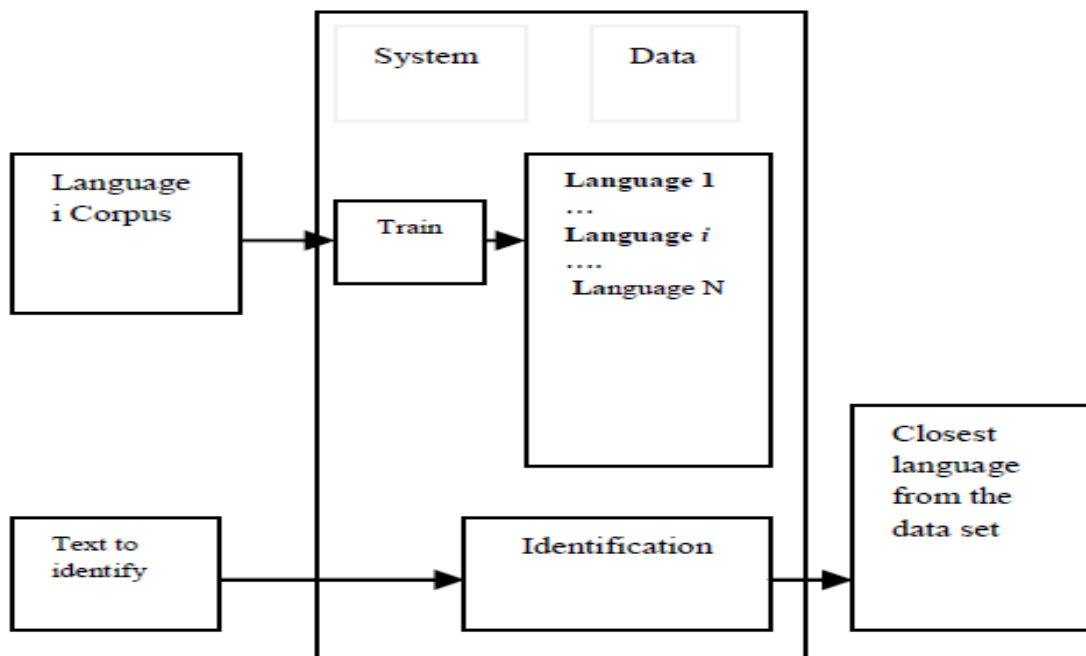


Figure 2.1: A general architecture of a language identifier based on computational approach

Numerous statistical approaches have been applied for text language identifications. Some of the prominent approaches include n-gram based models, Markov Models, Support vector machine, Bayesian classification etc. Several experimental studies reported high accuracy results for different collections of relatively long texts with proper grammar.

In this section, we discuss the different approaches that used to build models of languages that can be used to determine what language a document is written in. Parallel to the apparent diversity in document representation, there is a corresponding diversity in the descriptions of the learning algorithms applied to induce language classifiers.

2.6.1 Bayesian Classification

Number of the learning algorithms applied to LangID can be understood in the framework of Bayesian classifier, in which it computes $P(Li/D)$, the probability of a given language Li from a closed set of candidate languages L given a particular document D . The identified language l of document D is thus determined as the most likely language conditioned on the document D (Equation 2.1).

$$l = \underset{Li \in L}{\operatorname{argmax}} P(Li|D) \quad (2.1)$$

Bayes' theorem allows to re-express the likelihood of the language given the document ($P(Li/D)$) in terms of the product of the likelihood of the document given the language ($P(D/Li)$) and the prior probability of Li ($P(Li)$), normalized by the document probability $P(D)$ (Equation 2.2).

$$l = \underset{Li \in L}{\operatorname{argmax}} \frac{P(D|Li)P(Li)}{P(D)} \quad (2.2)$$

Since $P(D)$ is independent of Li , it does not affect the relative ordering of languages and thus can be dropped for purposes of determining the most likely language (Equation 2.3).

$$l = \underset{Li \in L}{\operatorname{argmax}} P(D|Li)P(Li) \quad (2.3)$$

To Implementing a Bayesian classifier require methods for estimating the likelihood of a document given a particular model of a language ($P(D/Li)$), as well as the prior probability over the set of languages ($P(Li)$). To estimate these two quantities methods are differ. Approaches to computing $P(D/Li)$ include Markov processes [21, 26], naive Bayes methods [27, 28, 29, 30, 31], and compressive models [32]. Language identifiers based on neural networks can also be

understood in this context, as each node in the output layer effectively computes the likelihood of the input under the class modeled by that particular node [34].

Where $P(L_i)$ is estimated, it is normally by maximum likelihood methods [30]. However, it is also common to assume a uniform prior [21, 27, 35, 28, and 32]. A uniform prior encodes the notion that no assumptions are made about what languages a document is most likely to be written in without seeing a document; it is considered to be equally likely that the document is written in any of the languages the classifier knows about. Depending on the application, this may or may not be a desirable characteristic of the classifier.

Another characteristic of Bayesian methods is that, under the assumption that the input document is written in a single language, it is possible to determine when sufficient evidence to make a decision has been collected and thus avoid processing the rest of the document [28, 33].

2.6.2 N-gram Method

A. Graph-based N-gram Method

Tromp and Pechenizkiy [36] describe a Language Identification Graph-based N-gram Approach (**LIGA**) for LangID. They use N-gram presences and occurrences and order, by creating a graph language model on labelled data. The weights of the nodes represent the frequencies of trigrams and the weights of the edges capture transitions from one character trigram to the next. To create a language model, the authors' use a training corpus of texts in that language. They calculate the frequencies of trigrams and their transitions and divide these counts by the total number of nodes or edges in the language.

For example In the word **lemon**, the nodes of the graph would be the trigrams: - _le, lem, emo, mon and on_, and the edges would be (_le,lem), (lem,emo), (emo,mon) and (mon,on_). In total, there are 5 trigrams and 4 transitions (edges) between them. Each trigram has a frequency of 1/5 and each transition has a frequency of 1/4.

If two sentences from different languages are taken, for example, “een test” in Dutch and “a test” in English, the resulting graph will be as shown in Figure 2.2 In this figure all nodes and edges

for these sentences are shown. It can be seen, that some nodes and transitions are shared between these two languages.

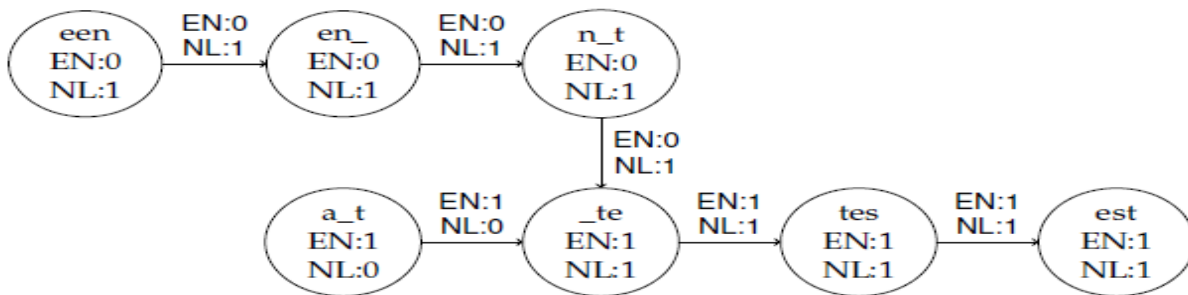


Figure 2.2: The graph resulting from the example training set

For example, if the training corpus contains the texts above, there would be 4 total trigrams in English and 6 in Dutch, 3 total edges in English and 5 in Dutch. To detect the language of the text “a tee”, a flat graph is made, as shown in Figure 2.3. For each language a so-called path matching score is computed. Only one node and no edges from the Dutch corpus are matched. On the other hand, two nodes and one edge from English corpus are matched. The path-matching score for Dutch is $1/6$ and for English $1/4 + 1/3 + 1/4 = 5/6$. If this is the highest score out of all the language models, the text “a tee” will be classified as English.

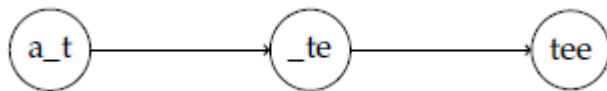


Figure 2.3: The graph resulting from the evaluation text

B. Rank-order Statistics Classifier

To determine the language of a document, Cavnar and Trenkle [24] use a technique that calculates a so called out-of-place measure for each N-gram of the document model. It determines the distance between an N-gram of the document model and the different language models. This technique is also called rank-order statistics. An example is shown in Figure 2.4.

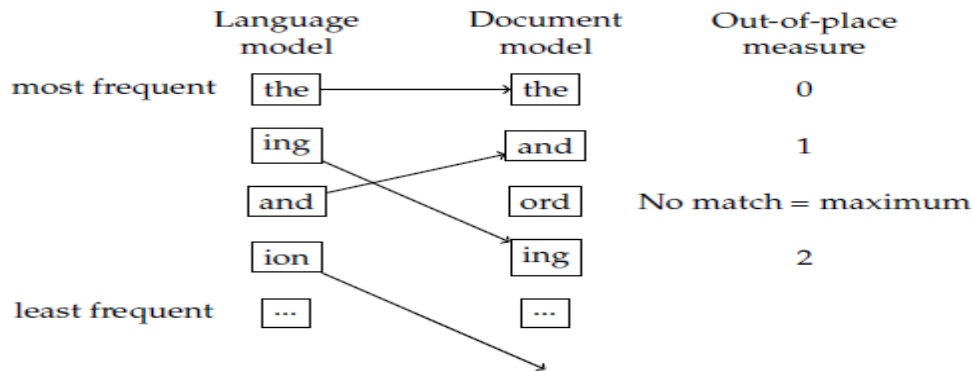


Figure 2.4: Example of the rank-order statistics classifier

If equal N-grams have the same rank in both models, like the N-gram “the”, distance between them is zero. If the respective ranks for equal N-grams vary, their distance is the number of ranks between them, so the distance between the N-grams “ing” is 2. If an N-gram from the document model, like the N-gram “ord”, is not found in the language model, their distance is defined as a maximum out-of-place value, which is generally the amount of N-grams in the language model. This is used to distinguish the correct language from the one with no matches. Subsequently, the sum of all out-of-place measures is the distance between the document model and the language model. Such distance is calculated for all languages and the smallest one indicates the language of the document.

2.6.3 Markov Model

Markov chain-based method was used for written language identification In Dat Tran and Dharmendra Sharma [37] work. By giving a training document in a specific language, each word can be represented as a Markov chain of letters. Using the entire training document regarded as a set of Markov chains, the set of initial and transition probabilities can be calculated and referred to as a Markov model for that language. Given an unknown language string, the maximum likelihood decision rule was used to identify language.

The main principles for a LangID system is that it should be fast for real-time processing, efficient, requires minimum storage, and robust against textual errors. A Markov chain-based method is proposed for language identification in [37] that satisfies this LangID principle. The occurrences of letters in a word can be regarded as a stochastic process and hence the word can be represented as a Markov chain where letters are states. The occurrence of the first letter in the

word is characterized by the initial probability of the Markov chain and the occurrence of the other letter given the occurrence of its previous letter is Markov Model characterized by the transition probability.

Given a text document in a specific language as a training set, the initial and transition probabilities for all Markov chains representing all words in the text document are calculated and the set of those probabilities is regarded as a Markov model for that language.

In order to identify language for an unknown string, the maximum likelihood decision rule was used. Words in the string are regarded as Markov chains and for each language model built in the training session, the initial and transition probabilities taken from the language model are used to calculate the probability of the unknown string for that language. The unknown string is then identified to the language that has the maximum probability. Implementation of the training and classification algorithms of Markov model is summarized as follows.

Given a training language document, it is first preprocessed to remove all special, common characters, and punctuation marks such as commas, columns, semi-columns, quotes, stops, exclamation marks, question marks, signs. The initial and transition probabilities are then calculated.

- Use the training sets of all languages to be identified, determine a common letter set containing M alphabetical letters.
- for each training language set, do the following
 - ✓ Remove all special characters to obtain the set of words X
 - ✓ Using all words in the set X , calculate the initial probabilities and the transition probabilities
 - ✓ Save all the probability values to a set λ and regard this set as the language model.
- Save the letter set for identification purpose.

Given an unknown language string, it is also preprocessed as shown in the training algorithm for the training documents. For each language model, the probability of the unknown string given the model is calculated. The maximum likelihood decision rule is used to identify language.

- Read all the language models and the letter set obtained from the training session.
- given an unknown language string, preprocess it to remove all special characters to obtain the set of words X
- for each language model, calculate the probability of the word set X given the language models
- the unknown string is then classified to the language that has the maximum probability

2.6.4 Support Vector Machine

For pattern classification Support vector machines have proven to be a powerful technique. SVMs map inputs into a high dimensional space and then separate classes with a hyper plane [38].

It is a kind of large-margin classifier and vector space based machine learning method. Where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data (possibly discounting some points as outliers or noise). SVMs are inherently two-class classifiers [39].

For text classification one needs to extend SVM to handle more than two classes. To achieve this most common technique in practice has been to build $|x|$ one-versus-rest classifiers (commonly referred to as “one-versus-all” or OVA classification), and to choose the class which classifies the test document with greatest margin. Another strategy is to build a set of one-versus-one classifiers, and to choose the class that is selected by the most classifiers. While this involves building $\frac{|x|(|x|-1)}{2}$ classifiers, the time for training classifiers may actually decrease, since the training data set for each classifier is much smaller [39].

It is possible constructing of multiclass SVMs as better alternative, by building a two-class classifier over a feature vector $\phi(\bar{x}, y)$ derived from the pair consisting of the input features and

the class of the datum. At test time, the classifier chooses the class $y = \operatorname{argmax} y' \overline{W}^T \phi(\vec{X}, y')$ [39].

SVM has been used successfully in many real-world problems; text categorization, image classification, bioinformatics (Protein classification, cancer classification), and hand written character recognition.

2.7 Non Computational Methods

Non-computational approaches requires the researchers to have sufficient knowledge about the language to be identified such as diacritics and symbols, stop words, character combinations etc. Sometimes this method is chosen because stop words, diacritics and other symbols are very specific to a language, although some languages have some similar words and special characters they are not all common. While computational approaches rely on statistical techniques rather than linguistic knowledge to solve related problems.

Stop words, from the point of view of LangID, are defined as the most frequent terms as determined by a representative language sample. This list of words has been shown to be effective for language identification because these terms are very specific to a language [40]. These words prove to be very effective for language identification. Although with different semantical meanings, stop words can be very similar, even the same, especially for related languages.

For example the word “አነ” is a stop word, which appears in Geez and Tigrigna but does not appear in Amharic. Other terms are very specific to a language and the scoring can be skipped altogether if they are found, e.g., the stop word “ነው” is only found in Amharic, “አዩ” is only found in Tigrigna, “ዐ.እቱ” is only found in Geez etc.

Diacritics also prove very useful for LangID. Some diacritics appear for certain language. If the analyzed text is written correctly, then, only by looking at the set of diacritics and the stop words,

a LangID method can accurately classify the given text and no scoring is necessary, especially in the cases where the diacritics are unique to the language and are widely used.

Even though, stop words usually are a good measure for language identification, for small and large texts the accuracy of language identification can be improved by using diacritics. In order to improve the accuracy and remove miss-classification the stop words dictionary must be well-built, conveniently by experts in the field.

2.8 Hybrid Approach

As language identification can be done by using the two approaches: Non-statistical approach and statistical approach. Non statistical approaches are basically linguistic approaches which require complete knowledge about the rules of language used. Statistical approaches are basically machine learning approaches which require less human efforts.so, by combining those two approaches we will increase the performance of our system. [41] Uses hybrid approach i.e. combination of linguistic and statistical approach. The feature set is derived from the linguistic knowledge that contains words and diacritics. These word and diacritics are transformed into relative frequency by using statistical approach. The vector space model has been selected as presentation model for presenting the transformed data [42]. Each training and testing data is presented by the document word frequency vector. This frequency vector is suitable for obtaining feature set of nine languages (English, French, German, Dutch, Italian, Portuguese, Turkish, Spanish, and Swedish). Weighing factor is also used for increasing performance. The word which passes in more languages will have small weighing factor. Besides it three classification algorithms are used: SVM for classification, MLP (Multilayer Perception), LDA (Linear Discriminant Analysis). SVM is a kernel based classification algorithm [43], MLP is the neural net based algorithm [44], and LDA is a statistical based classifier [45].

2.9 Empirical Evaluation

The most common approach is to treat the task as a document-level classification problem. Given a set of evaluation documents, each having a known correct label from a closed set of labels (often referred to as the “gold-standard”), and a predicted label for each document from the same set, the document-level accuracy is the proportion of documents that are correctly

labeled over the entire evaluation collection. This is the most often-reported metric, and conveys the same information as the error rate, which is simply the proportion of documents that are incorrectly labeled (i.e. 1 - accuracy).

Authors sometimes provide a per-language breakdown of results. There are two distinct ways in which results are generally summarized per-language: (1) precision, in which documents are grouped according to their predicted language; and (2) recall, in which documents are grouped according to what language they are actually written in. More formally, consider a set of documents $D = \{D_1 \dots D_m\}$ and a set of languages $L = \{L_1 \dots L_n\}$. For each document D_x we denote that the document is written in language L_y by $D_x \rightarrow L_y$, and that the system predicts the document is written in L_z by $D_x \triangleright L_z$. We use an over line to denote negation, for example $D_x \rightarrow \bar{L}_y$ denotes that D_x is not written in L_y . For each language $L_i \in L$, each document can fall into four possible categories:

True Positive (TP) $D_x \rightarrow L_i$ and $D_x \triangleright L_i$

False Positive (FP) $D_x \rightarrow \bar{L}_i$ and $D_x \triangleright L_i$

False Negative (FN) $D_x \rightarrow L_i$ and $D_x \triangleright \bar{L}_i$

True Negative (TN) $D_x \rightarrow \bar{L}_i$ and $D_x \triangleright \bar{L}_i$

Given a gold-standard and a set of predictions, the frequency of each category can be tabulated for each language. On the basis of these counts, precision (P) and recall (R) are defined as the following ratio of counts:

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN}$$

Earlier example has tended to only provide a breakdown; based on the correct label (i.e. only reporting per-language recall). This gives us a sense of how likely a document in any given language is to be classified correctly, but does not give an indication of how likely a prediction for a given language is of being correct. Under the monolingual assumption (i.e. each document is written in exactly 1 language), this is not too much of a problem, as any false negative for one language must also be a false positive for another language, so precision and recall are closely linked. Nonetheless, later authors have tended to explicitly state both precision and recall for

clarity. It is also common practice to report an F-score (F), which is the harmonic mean of precision and recall:

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

The F-score (also sometimes called F-measure) was developed in information retrieval to measure the effectiveness of retrieval with respect to a user who attaches different relative importance to precision and recall [46]. When used as an evaluation metric for classification tasks, it is common to place equal weight on precision and recall, and this has also been the practice in work to date on LangID that has used the F-score [47, 48, 49, and 50].

In addition to evaluating performance for each individual language, authors have also sought to convey the relationship between classification errors and specific sets of languages. Errors in LangID systems are generally not random; rather, certain sets of languages are much more likely to be confused. For example, [27] found that Norwegian documents had an elevated chance of being misclassified as Swedish, compared to a range of other European languages. [51] Found an elevated chance of misclassification between Croatian, Serbian and Slovenian, and this specific set of languages has been the focus of later research [52]. The typical method of conveying this information is through the use of a confusion matrix, a tabulation of the distribution of (predicted language, actual language) pairs. Confusion matrices can be presented over the entire language set [51], or can be cropped to focus on a particular subset of languages [35].

Presenting full confusion matrices becomes problematic as the number of languages considered increases, and as a result has become relatively uncommon in work that covers a broader range of languages. Per-language results are also harder to interpret as the number of languages increases, and so it is common to present only collection-level summary statistics. There are two methods to summarize across a whole collection: (1) giving each document equal weight; and (2) giving each class (i.e. language) equal weight. (1) referred to as a micro-average, and (2) as a macro average. For LangID under the monolingual assumption, micro-averaged precision and recall are the same, since each instance of a false positive for one language must also be a false negative for another language. In other words, micro-averaged precision and recall are both simply the collection-level accuracy. On the other hand, macro-averaged precision and recall give equal weight to each language. In datasets where the number of documents per language is

the same, this again works out to being the collection-level average. However, LangID research has frequently dealt with datasets where there is a substantial skew between classes. In such cases, the collection-level accuracy is strongly biased towards more heavily-represented languages.

There are two possible methods to calculate the macro-averaged F-score. The first is to calculate it as the harmonic mean of the macro-averaged precision and recall, and the second is to calculate it as the arithmetic mean of the per-class F-score.

2.10 Multilingual Documents

Multilingual documents are documents that contain text in more than one language. Recent research has investigated how to make use of multilingual documents from sources such as web crawls, [54], forum posts [55] and microblog messages [56]. However, most LangID methods assume that a document contains text from a single language, and so are not directly applicable to multilingual documents. As a result, research to date has sometimes discarded multilingual documents before carrying out experiments [24, 57].

Handling of multilingual documents has been named as an open research question [2]. Most natural language processing techniques presuppose monolingual input data, so inclusion of data in foreign languages introduces noise, and can degrade the performance of NLP systems [59, 60]. Automatic detection of multilingual documents can be used as a pre-filtering step to improve the quality of input data. Detecting multilingual documents is also important for acquiring linguistic data from the web [61, 62], and has applications in mining bilingual texts for statistical machine translation from online resources [63, 64, 56], or to study code-switching phenomena in online communications [55]. There is also been interest in extracting text resources for low-density languages from multilingual web pages containing both the low-density language and another language such as English [65, 54].

The need to handle multilingual documents has prompted researchers to revisit the granularity of LangID. Many researchers consider document-level LangID to be relatively easy [22], and that sentence-level [66] and word-level [54, 55] LangID are more suitable targets for further research.

However, word-level and sentence-level tokenization's are not language-independent tasks, and for some languages are substantially harder than others [67]. Furthermore, reducing the granularity of LangID also presents challenges in dealing with shorter quantities of text on which to base the prediction.

Handling multilingual documents is to attempt to segment them into contiguous monolingual segments. In addition to identifying the languages present, this requires identifying the locations of boundaries in the text which mark the transition from one language to another.

In general the solution to the multilingual identification problem completely depends on the following assumptions [68]:-

(I). Diversity Assumption: The accuracy of a language identifier depends on the number of languages from which the identifier has to select one. This reflects the coverage of the identifier in terms of linguistic diversity, which implies an assumption about linguistic diversity. There are two kinds of diversity assumptions, both of which can be applicable at the same time.

- **Global Diversity Assumption:** This is about how many languages are assumed to be in the world. In practical terms, this is reflected in the number of languages for which the system has been trained.
- **Local Diversity Assumption:** For a particular user or for a particular context, the number of possible and relevant languages may be less than the number for which the system has been trained. For example, a user may only be interested in the documents in European languages, even though the system has been trained for languages from around the world. In such a case, a local diversity assumption is likely to increase the accuracy and speed of the identifier.

(II). Limited Ambiguity Assumption: Multilingual documents have text in more than one language, but if we do not assume a small limit to this number, the problem may not be tractable, unless we assume large segment sizes. All the algorithms for monolingual identification work well only when the test data size is sufficient, e.g., 100 characters. Thus, to make the problem solvable, we will make the limited ambiguity assumption, that the number of languages to be

disambiguated for a segment is small in number. In our experiments, we have assumed this number to be one, two, three or four, which means that the multilingual documents can be monolingual, bilingual, trilingual or Quadra lingual. Unlike the diversity assumption, this assumption is about the possible languages in a document, not in the world. Therefore, it applies for both a multilingual identifier and monolingual identifier.

(III). Language Switching Assumption: Another assumption that applies only to a multilingual identifier is the language switching assumption. This specifies how frequently or where a shift from one language to another can occur in a document. There are two such assumptions, only one of which can apply at a time.

- **Long Sequence Assumption:** This assumption says that the minimum segment size in any language is large enough for a monolingual identifier to identify its language accurately. If we make this assumption, the problem of segment identification actually becomes a problem of identifying where language shift occurs and from which language to which language. This is, of course, a less realistic assumption.
- **Isolated Word Assumption:** The more realistic assumption is that every word in the document can be in a different language, subject to the limited ambiguity assumption, i.e., language switch can occur at any word boundary.

In this thesis we adopt an isolated word assumption in order to make the proposed language identifier approach more realistic, in which every word of the document is labelled to a language category. In other word, the language switching is assumed to be occurring at a minimum unit of word level.

2.11 Closely-related Languages

Closely-related languages are a known problem for existing language identifiers [69, 70, and 71]. [16] Find that LangID methods are not competitive with word-based methods in distinguishing between national varieties of English. [69] Reports that, a character trigram model is able to distinguish Malay/Indonesian from English, French, German and Dutch, but handcrafted rules are needed to distinguish between Malay and Indonesian. One kind of rule is the use of

“exclusive words” that are known to occur in only one of the languages. A similar idea is used by [53], which automatically learn a “blacklist” of words that have a strong negative correlation with a language – i.e. their presence implies that the text is *not* written in a particular language. [66] Also adopts such “discriminative training” to make use of negative evidence in LangID.

[71] Investigated the issue of document representation for closely related languages, since typical LangID approaches use a character n -gram representation of text, but recent work on closely-related languages seems to favor word-based representations [73, 52, and 72], comparing n -gram based representations to bag-of-words representations for LangID over varieties of Spanish, Portuguese and French. The results were inconclusive, with word-level models being better for Spanish and character n -gram models being better for Portuguese and French.

2.12 Ethiopian Semitic language

In Ethiopia there are 83 different languages with up to 200 different dialects spoken. It divided into four major language groups. These are Semitic, Cushitic, Omotic, and Nilo-Saharan. The Semitic languages are spoken in northern, central and eastern Ethiopia (mainly in Tigray, Amhara, Harar and northern part of the Southern Peoples' State regions). They use the Ge'ez script which is unique to the country; it consists of 33 letters, each of which denotes 7 characters, making a total of 231 characters [4].

Semitic languages share common characteristic features [91, 93]. They use a special writing system called the Ge'ez or Ethiopic alphabet. The normal syllable is considered to be a consonant followed by a vowel. If a consonant ends a syllable, the sixth, neutral vowel is used with it. Most consonants are written in seven different forms corresponding to the seven different vowels [91, 92, and 93].

The Semitic Languages are: Adarigna, Amharigna, Argobba, Birale, Gafat, Ge'ez, Guragigna, Chaha group (Chaha, Muher, Ezha, Gumer, and Gura), Inor group (Inor, Enner, Endegegna, Gyeto, and Mesemes), Silt'e group (Silt'e, Ulbareg, Enneqor, and Walane), Soddo group (Soddo, Gogot, and Galila), Tigrigna, and Zay [4].

Amharic is the second most spoken Semitic language in the world, after Arabic, and the official working language of Ethiopia [5, 6]. The 2007 census counted nearly 22 million native and 15 million secondary speakers in Ethiopia. Amharic is spoken by 3 million emigrants outside Ethiopia. Most of the Ethiopian Jewish communities in Ethiopia and Israel speak Amharic. In Washington DC, Amharic became one of the six non-English languages in the Language Access Act of 2004, which allows government services and education in Amharic. [7] Furthermore, Amharic is considered as a holy language by the Rastafarian religion and it is widely used among its followers worldwide.

Tigrigna is mainly spoken in northern Ethiopia and Eritrea, with around 6,915,000 total speakers. Tigrinya speakers in Ethiopia are around 4,320,000 in the northern Tigray Region. In Eritrea are approximately 2,540,000 in the southern and central areas. It is also spoken by emigrants from these regions, including some Beta Israel [8].

Gurage languages (also known as Guragie) [15] are a group of South Ethiopia languages, which belong to the Semitic branch of the Afro-asiatic family. They are spoken by the Gurage people, who inhabit the Gurage Zone within the larger multi-ethnic Southern Nations, Nationalities, and Peoples Region in south western Ethiopia.

Generally, Guragigna has 3 major categories: Northern, Eastern and Western Gurage [8][15].

In the Northern group

- Soddo (Kistane): speaks in Oromia region and Southern Nations, Nationalities, and Peoples' region. It has 255,000 native and 60,500 secondary speakers (1994 census).

In the Eastern group

- Silt'e : speaks in Southern Nations, Nationalities, and Peoples' region. It has 935,000 (2007 census) native speakers.
- Wolane: speaks in Oromia region and Southern Nations, Nationalities, and Peoples' region.
- Zay (Zway): speaks in Oromia region: Lake Zway shores and east islands. It has 4,880 (1994 SIL), native speakers.

In the Western group

- Inor: speaks in Southern Nations, Nationalities, and Peoples' region. It has 280,000 native speakers
- Mesmes : speaks in Southern Nations, Nationalities, and Peoples' region.
- Mesqan: speaks in Southern Nations, Nationalities, and Peoples' region. It has 195,000 (2007 SIL) native speakers.
- Sebat Bet Gurage: speaks in Southern Nations, Nationalities, and Peoples' region. It has 1,480,000 (2010 UNSD) native speakers.

However, in this investigation due to availability of language corpus Sebat Bet Gurage textual documents is used for language identification.

Ge'ez was the official language of the kingdom of Aksum and Ethiopia imperial court. Today, it remains only in the Ethiopia Orthodox Tewahedo church, the Eritrean Orthodox Tewahedo church, the Ethiopia Catholic church, and the Beta Israel Jewish community. Tigrigna, Guragigna and Amharigna (Amharic) are the modern languages which are derived from Ge'ez [14].

As explained earlier, the families of Ethiopian Sematic languages are much closed languages, which have usually common sequence of character and also have words in common. For instance, the word “ወሰን” is common word for Amharic, Geez and Tigrigna. Hence, in order to identify the language category of such common word an approach which can disambiguate such ambiguity is required. In our proposed approach we used the contextual information of a word in order to disambiguate the language category of a word.

2.13 Summary

This chapter begins with brief discussion of language identification as one of the NLP applications and its processes of labeling the language category for a given document. I also indicate the useful application areas of language identification such as Multilingual Information Retrieval (MLIR), text mining, Knowledge Discovery in Text (KDT), text to speech applications and so on.

As well in this chapter, we have also discussed about the language feature sets which are used for automatic language identification task and these are either words or characters. In addition, we also briefly presented approaches which are able to use these language feature sets in to identify the language of given input document and through reviewing different papers , we classify these approaches into three basic categories i.e. computational approach , non-computational approach and hybrid of these two approaches. An explanation of these language identification approaches also made briefly.

On the other hand, we also presented about the evaluation techniques of language identification i.e. recall, precision and F-measure. Beside this, we also briefly discussed about the issue of multilingual language identification, identify the language of documents written with two or more languages. Finally, the problem and issues of language identification tasks for much related languages also presented.

Chapter 3

Related Work

3.1 Introduction

Research in the area of language identification has grown steadily over the years. However, most researchers have concentrated attention on English and the other European languages for obvious reasons. Since English was the original language of most computer designers and users, it became like the official language of computer usage. Naturally, the spread of computer use again flowed first among the European languages, and the most pressing issues then were how information exchange among these languages could be facilitated. Thus, for many years research on language identification and other areas of natural language processing concentrated on the areas of European and later also on the Asian languages. Only recently has there been some interest in expanding the coverage in terms of other languages. Africa has been particularly neglected in the area of language identification research. Indeed, only in 2006 the first African language was featured in any language identification research. In general the coverage of language identification research on the languages of the world has also been low. According to [73], more than 7000 languages are listed in the Ethnologies as living languages spoken on earth. However, most of the published research on language identification focuses on languages that are spoken by large numbers of speakers and are also well resourced in terms of written language resources or both [2]. The most important reason for the omission of resource-poor languages lies mainly in the fact that the most popular identification techniques are statistical in nature, and these require large amounts of data to build the necessary evaluation models. This situation is bound to change with the development of techniques like the spelling checker method used by [15] which is suitable for the identification of under-resourced languages. Such a development will contribute greatly in reducing the negative effects of the language digital divide.

Although language identification is often portrayed as a solved problem [22], much research is still going on in this area because there are yet outstanding issues, including the identification of minority languages, open-class language identification, sparse or impoverished training data,

language identification of multilingual documents, standard corpora, and the effects of pre-processing and encoding standards [74, 2].

In this chapter, the most related researches and approaches in language identification are summarized. Due to the long research history of this area it is difficult to give a comprehensive overview of the most important ideas. But we will try to see works that are related with our work by dividing in to sections i.e. works on monolingual, multilingual and works on closely related language.

3.2 Language Identification for Monolingual

The dominant approach in the literature of monolingual text is the character-based n-gram model. Cavnar and Trenkle [24] used the n-gram profile, based on the most frequent character n-grams in a text. In their work training sets are in 8 languages on the order of 20K to 120K bytes in length have been used. Their validation set consisted of 3478 articles from a newsgroup hierarchy of Usenet that were fairly pure samples of a single language. From these articles, punctuation marks were deleted. Words were tokenized and delimited by white space before and after.

Cavnar and Trenkle kept track if an article was over or under 300 bytes in length and varied the number of the N-gram frequencies from 100 to 400. The average text size was 1700 bytes. As shown in Table 3.2, the article length had a minor impact on the overall results of the language identification compared to the number of N-gram frequencies. Overall, their system showed the best performance at a training language model length of 400 N-grams, misclassifying only 7 articles out of 3478 and having an overall classification rate of 99.8%.

Article length (bytes)	< 300	< 300	< 300	< 300	> 300	> 300	> 300	> 300
Training model length (N-grams)	100	200	300	400	100	200	300	400
Overall correct	92.9%	97.6%	98.6%	98.3%	97.2%	99.5%	99.8%	99.8%

Table 3.2: Comparison of the results using article and training model lengths Cavnar and Trenkle work

They had also found interesting anomalies. An increasing N-gram model length decreased the percentage of correctly detected languages. This was mainly, due to the multiple languages that had a similar distance measures from the tested article.

Grefenstette [27] compared the short words approach with trigrams. The author used one million characters of text from the European Corpus Initiative (ECI) collection 2 and considered the following ten languages: Danish, Dutch, English, French, German, Italian, Norwegian, Portuguese, Spanish and Swedish. The author tokenized the sentences and counted all words and trigrams occurrences. The researcher took words that have a length of five or less characters. Moreover, punctuation marks were not removed before generating the Ngram- based language models, which resulted in N-grams that contain only commas or dots. Each language was characterized with trigrams appearing at least 100 times of amount resulting from 2550 to 3560 N-grams and with words that occur at least three times resulting from 980 to 2750 words, depending on the language. On test strings with 1 to 5 words, the results of the short words approach were worse, because of the high probability that no word is found in the language model. But with at least 15 words in test string round 99.9% of strings were correctly recognized. The trigram approach has shown better results than the short words approach almost in all of his tests. The samples with more words performed better, but starting with 15 words all methods performed equally well with round 99.9%.

Similarly, Prager [75] used to generate a training set from 100 Kbytes of text of 13 Western European languages. These languages share etymological roots and have largely overlapping character sets, what made the task more difficult. As a validation set, he took chunks of text with sizes from 20 to 1000 bytes. The researcher compared the results for all sizes of chunks, tried to find the N-gram length and performed additional experiments using both N-grams and words together as features. When they were used together, character sequences recognized both as a word and an N-gram were treated solely as a word in both indexing and matching processes.

Unlike Grefenstette, Prager used words up to four letters for the short words approach and calls them “stop-words”. The researcher noted that words of unrestricted length did better than short words, as he had expected, but both of them had good performance. The researcher also thought

that a set of only function words, such as pronouns, prepositions, articles and auxiliaries tend to be quite distinctive, and it should perform as good as a set of short words, but actual lists of such words were not available to Prager. The combination of short words and N-grams showed better results than either of these methods alone. Quad grams and words of unrestricted length had the best performance. The best N-gram length was 4, followed by 5, 3 and 2, which performed poor on the small sizes of chunks. As it was predicted by Prager, the longer input text was better recognized than the small ones. Prager correctly identified 83.6% of 20-character test texts.

Furthermore, Kheireddine A. et al [11] Use the fusion of two algorithms i.e. CBA and WBA for automatic language identification of noisy texts in 32 languages. Those algorithms are executed in parallel (at the same time), and once the two processes are finished without classifying the language, add the sum of frequencies of the two algorithms for each language. $Sum = freqCBA_i + freqWBA_i$ Where $freqCBA_i$ is the sum of character frequencies in language i , and $freqWBA_i$ is the sum of word frequencies in language i . Finally, classification of the text will be according to the language that has the highest sum of frequencies. They evaluate the efficiency of their approaches on a small number of languages and on large number of languages 10 and 32 different languages respectively. The identification score of WBA is 90% in the two tests. Here; all texts are recognized correctly, but Chinese texts, are not recognized. In their research work, they proposed three basic language identification algorithms: characters based identification (CBA), special character based identification (SCA) and common words based identification (WBA). Furthermore, they proposed two hybrid approaches based on the combination of the two previous methods (character based identification and common words based identification): HA1 (a sequential combination) and HA2 (uses a parallel fusion). These two combinations have presented good performances, especially the parallel fusion based approach (HA2), which got an identification score of 100% with 10 languages and a score of 97.78% with 30 languages. The results of HA2 are better than those obtained by HA1, which shows that the parallel fusion is quite interesting.

3.3 Language Identification for Multilingual

Research to date on LangID for multilingual documents has been limited. Linguini (Prager 1999) is a language identifier that supports identification of multilingual documents. The system is based on a vector space model, and cosine similarity between a feature vector for the test document and a feature vector for each language L_i , computed as the sum of feature vectors for all the documents for language L_i in the training data. The elements in the feature vectors are frequency counts over byte n -grams ($2 \leq n \leq 5$) and words. Language identification for multilingual documents is performed through the use of virtual mixed languages. Prager [75] shows how to construct vectors representative of particular combinations of languages independent of the relative proportions, and proposes a method for choosing combinations of languages to consider for any given document. One weakness of this approach is that for exhaustive coverage, this method is factorial in the number of languages, and as such intractable for a large set of languages. Furthermore, calculating the parameters for the virtual mixed languages becomes unfeasibly complex for mixtures of more than 3 languages.

Teahan [32], proposed a system based on text compression that identifies multilingual documents by first segmenting the text into monolingual blocks. Mandl *et al.* [81] detect “language shift” using an eight-word LangID window.

Rehurek and Kolkus [17], describe and evaluate an algorithm that segments input text into Monolingual blocks, perform language segmentation by computing a relevance score between terms and languages, smoothing across adjoining terms and finally identifying points of transition between high and low relevance, which are interpreted as boundaries between languages.

Yamaguchi and Tanaka-Ishii [65] use a minimum description length approach, embedding a compressive model to compute the description length of text segments in each language. They present a linear-time dynamic programming solution to optimize the location of segment boundaries and language labels. Their data was artificially created by randomly sampling and concatenating text segments (40-160 characters) from monolingual texts. Therefore, the language switches do not reflect realistic switches as they occur in natural texts. [65] Observe that in

trying to gather linguistic data for “non-major” languages from the web, one challenge faced is that documents retrieved often contain sections in another language. SEGLANG (the solution of [65]) concurrently detects multilingual documents and segments them by language, but the approach is computationally expensive and has a tendency to over-label.

Closely related to the idea of text segmentation by language is the idea of word-level LangID [54 and 55]. Here, the task becomes to label each word in the document with a specific language. Work to date in this area has assumed that word tokenization can be carried out on the basis of whitespace, and that the languages present in the document are known in advance.

King and Abney [54] make use of conditional random fields, and introduce a technique to estimate the parameters using only monolingual data, an important consideration as there is no readily-available collection of manually-labeled multilingual documents with word-level annotations. The solution of [54] is incomplete, and they specifically mention the need for an automatic method “to examine a multilingual document, and with high accuracy, list the languages that are present in the document”.

Nguyen and Dogruoz [55] present a two-pass approach to processing Turkish-Dutch bilingual documents, where the first pass labels each word independently and the second pass uses the local context of a word to further refine the predictions. They achieve an accuracy of 98%. Their results reveal that language models are more robust than dictionaries and adding context improves the performance. They evaluate their methods from different perspectives based on how language identification at word level can be used to analyze multilingual data. The highly informal spellings in online environments pose challenges.

MarcoLui et al., [3] presents a system for language identification in multilingual documents using a generative mixture model inspired by supervised topic modeling algorithms, combined with a document representation for monolingual documents. The results illustrates that the proposed system outperforms alternative approaches from the literature on synthetic data, as well as on real-world data from related research on linguistic corpus creation for low density languages using the web as a resource. They use Ben King [54] data, which consists of 149

documents containing 42 languages retrieved from the web using a set of targeted queries for low-density languages and it was manually annotated.

3.4 Language Identification for Closed-related Language

Language identification for closely-related languages has been studied for several languages like: Malay- Indonesian (Ranaivo-Malancon 2006), Indian languages (Murthy and Kumar 2006[47]), Serbo-Croatian languages (Ljubešić et al. 2007[52]; Tiedemann and Ljubešić 2012[53]), Australian- British-Canadian English (Lui and Cook 2013[16]), Belgian-Netherlandic Dutch (Peirsman et al. 2010[76]), Dutch dialects (Trieschnigg et al. 2010[77]), Mainland-Singapore-Taiwan Chinese (Huang and Lee 2008[58]), European-Brazilian Portuguese (Zampieri et al. 2012[82]), Spanish varieties (Zampieri et al. 2013[71]), French varieties (Diwersy et al. 2014[78]), and Arabic dialects (Elfardy and Diab 2013[79]; Zaidan and Callison-Burch 2014[80]). However, as far as the researcher knowledge, there is not researcher work is done for very closely related Ethiopian languages before.

Ranaivo-Malancon [69]; report that a character trigram model is able to distinguish Malay/Indonesian from English, French, German and Dutch, but handcrafted rules are needed to distinguish between Malay and Indonesian. One kind of rule is the use of “exclusive words” that are known to occur in only one of the languages.

A similar idea is used by Tiedemann and Ljubešić [53], propose two token-based approaches, one based on a Naive Bayes classifier and one based on weighted lists of blacklisted words, which automatically learn a “blacklist” of words that have a strong negative correlation with a language – i.e. their presence implies that the text is not written in a particular language. Both perform very well and significantly outperform state-of-the-art approaches to language identification. Based on their experiments a Naive Bayes model performs better for smaller amounts of data but highly depends on the comparability of the language data it is trained on. The blacklist approach is similar in essence but includes heavy feature selection. This leads to a larger generalization of the model and makes it perform better on less parallel data sets. The overall performance of the blacklist approach is also higher given the entire data set they train on

and improves the best baseline created using public language identification tools accuracy. Tiedemann and Ljubešić [53] report an overall accuracy of 97.7% on Bosnian/Serbian/Croatian, compared to 45% attained by TextCat.

Zampieri [71] investigated the issue of document representation for closely related languages, since typical LangID approaches use a character n-gram representation of text, but recent work on closely-related languages seems to favor word-based representations (Huang and Lee [58]; Tiedemann and Ljubešić [53]; Lui and Cook [16]), comparing n-gram based representations to bag-of-words representations for LangID over varieties of Spanish, Portuguese and French. The results were inconclusive, with word-level models being better for Spanish and character n-gram models being better for Portuguese and French.

Sreejith C et al [10] have proposed an N gram based approach for distinguishing between Hindi and Sanskrit texts which have a common script. They have used character based Ngram (unigram, bigram and trigram) training profiles and have achieved 99% accuracy. Beside this, character based Ngram is also good approach for language which has not word boundary like Japanese, Chinese etc.

3.5 Summary

This chapter reviewed different researches attempt to develop language identification systems for various languages. The review showed that there are various approaches can be followed for language identification. Since, various languages have different in their nature it is difficult to apply one algorithm for those language so, the approach may depend on the nature of the language.

From the literature review and related works it has been observed that various approaches can be followed in a text based language identification systems. A pure linguistic approach would be the best candidate where high classification accuracies are desired. Though these models would describe the language best, a large amount of linguistic expertise is required. Where collecting such knowledge is difficult statistical approach is a possible alternative. Statistical language models can be built from the statistics of words, letters or n-grams.

Even if LangID has great advantage for different NLP application, as we see on the related works Ethiopian sematic language is totally neglected so, doing higher NLP application for those language face challenge. This research is the 1st research for those languages. Hence the proposed work promises to overcome these challenges and give a direction for researchers who are interested on this area.

Most researcher used character based Ngram approach for identifying monolingual text However, in much closed languages like Amharic, Tigrigna, Ge'ez etc. this approach may not always work because occurrence of characters in some cases are similar. Also words which has small in character size like one or two is difficult to identify the language of text document with character based ngram approach only. So in order to alleviate this problem in the proposed work we use the best future of character ngram such as character ngram with fixed size that is commonly used n-grams $2 \leq n \leq 5$ and character ngram with infinite size, which means that the size is not limited in specific number rather it is based on the length of the word. This is a new approach for LangID, We will discuss in design and implementation part the detail of this and how it works.

Beside this , in both character ngram variation form we used the location feature set in order to enhance the capability of proposed language identifier, which is identify one ngram character feature in which location of the word is located. This technique is good for those languages which have many common character sequences and differ in some sequences.

In case of multilingual LangID there are different approaches which are applied before by different researcher. They tries to identify the number of languages by using different approaches as we discussed in prewise.

In order to develop multilingual language identifier, some researcher's use character window size. This method, take some fixed size of characters from the text and tries to predict the language of that fixed size characters. Tacking random size cannot show exact switching point of the language, since all words have not equal character window size and there is a problem of labelling a different character portion of a word in to different language category. However, the proposed approach identifies a language category at word level, and such feature handles the labelling of a character portion of a word in to different language category.

In addition to this, identifying a language category at word level is the best solution to detect the exact switching point of a language for multilingual closely related language documents. In order to capture the switching point of language category, the proposed approach determines the actual position of each word in the actual text of a document.

Moreover, to increase the efficiency of this approach, in the proposed approach used an optimization technique, which is the language profile in both character ngram as well as word ngram contents are organized by word length as index and this is a onetime process, once the vocabulary is indexed in this way it is only updated as and when necessary. Using such word length strategy is reducing the search time.

Since we work on multilingual closed related languages a word may found in more than one language which is common for two and above, so the system may face problem to which that word belongs, according to the context such problem needs additional approaches. So to overcome this problem we use contextual based approach which used to disambiguate the word based on different rules which is follow to disambiguate. Hence, the proposed approach can disambiguate for such language category ambiguity of a word based on words contextual information.

In general Language identification, as the task of determining the language a given text is written in, has progressed substantially in recent decades. However, three key issues posited in the literature [63,24,69] and that, as of today, cannot be considered solved include: (i) distinguishing similar languages [76], (ii) dealing with multilingual documents [43], and (iii) language identification for short texts [6,10,35,20,70,52].

This investigation aims to solve such unsolved problems of the language identification through the proposed general purpose language identification approach. The proposed approach can classify the textual document at any level (i.e. word, phrase, sentence and document) for both monolingual as well as multilingual setting.

Chapter Four

Design of Proposed Language Identifier

4.1 Introduction

As we explained earlier, identifying language of a text is an important component of a natural language processing (NLP). It used as preprocessing step for multilingual NLP application such as Machine translation, Part of Speech tagging, search engines information extraction, e-mail routing and filtering engines, text mining applications, identification of the language or encoding of WWW pages, information retrieval systems etc. In this chapter, a detail description of the developmental process of the proposed general purpose language identifier is described.

4.2 Design Goals

In this investigation, we are aim to design and implement an approach which can identify a language label of each word in a text that was written. First, we assume that a particular document to be identified contains one or more languages used in the corpus: Amharic, Tigrigna, Geez and Guragigna. This means that each word in the text receives one of these four labels.

The input test document given can contain only a single word, phrase, sentences or collection of sentences and can be written in one or more languages in a set of specified domain languages. In general the aim of this study is to develop a general purpose language identifier. For instance, if the following sentences are input into the proposed language identifier approach, the expected output would be Geeze, Tigrigna, and Amharic respectively.

Sentence 1: ከንቱ ውእቱ ዝንቱ ዐለም : :

Sentence 2: እዚ ዓለም ከንቱ እዩ : :

Sentence 3: ይህ ዓለም ከንቱ ነው : :

4.3 Architecture of General Purpose Language Identifier

As mentioned earlier, the goal of this study is to develop a general purpose language identifier , a language identifier which can identify the language of textual documents written in one language (i.e. monolingual textual documents) as well as textual documents written in more than one language (i.e. multilingual textual documents). As shown in Figure 4.1, the proposed general architecture of our general purpose language identifier is structured into two main phases i.e. training phase and testing phase. Each of these phases has its own sub proses as well as shared sub modules. In this section we will discuss design as well as implementation issue of each and every sub proses of these phases in detail.

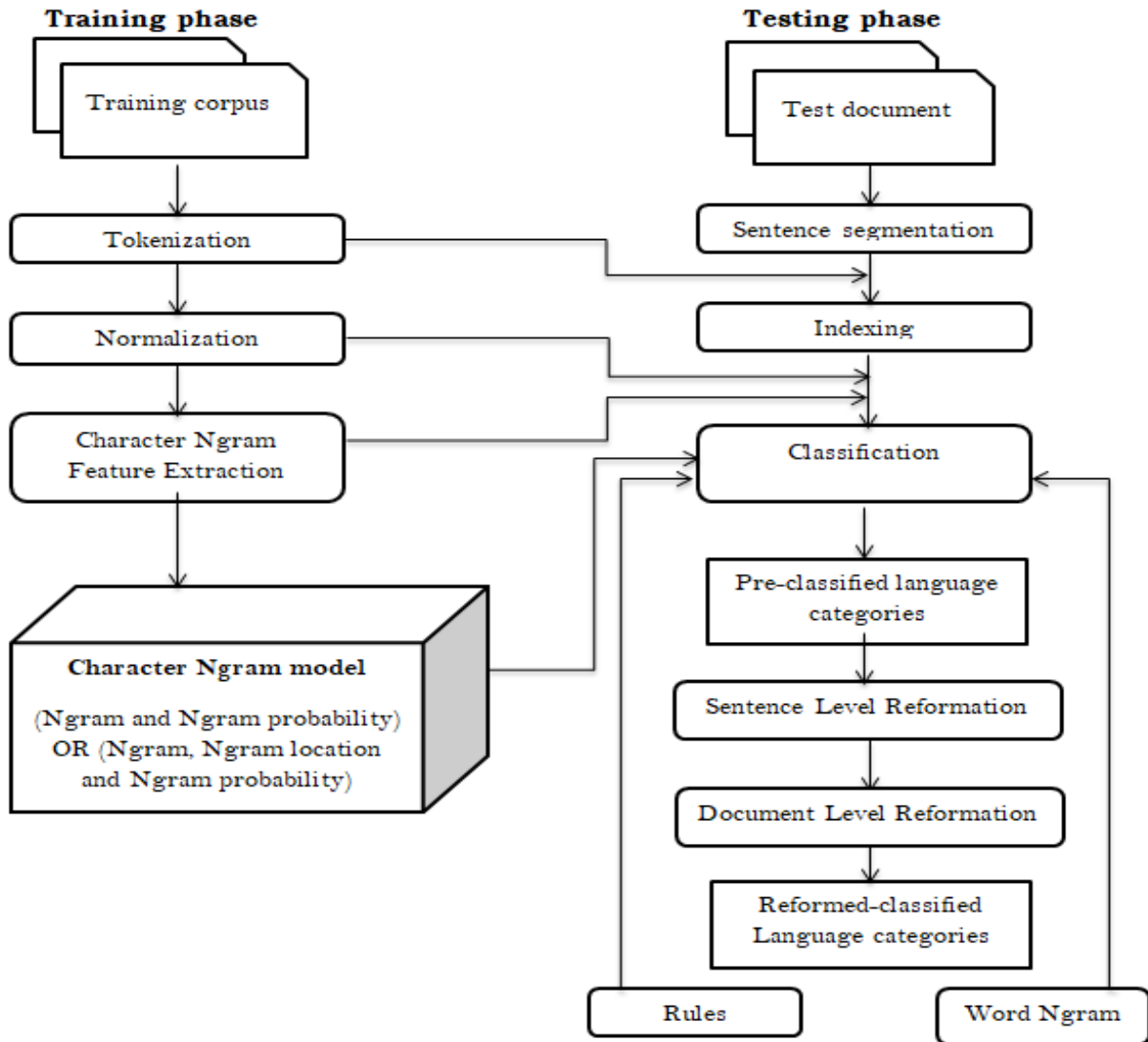


Figure 4.1: General Architecture of Proposed Language Identifier

Figure 4.1 illustrates that the input text document is given to system along the testing phase to be labelled to one or more language categories by proposed language identifier. The sentence segmentation splits the content of a document into bag of sentence and requests the tokenization module in order to split all bag of sentence into bag of words for further processes. During these operations the index information for each of segmented sentences as well as tokenized words are recorded. Then the normalization module is devoted in order to normalize the homophone characters found in a bag of words. After that, the classification module accepts indexed bag of words and returns a language label of each words in bag of words, the classification is done with a combination of rules, word ngram, character ngram approaches. From the classification module a language category for each word in bag of indexed words is provided and this is called pre-classified language categories. However, due to very similarity of the languages to be identified there is a probability of incorrectness during language assignment and in order to perform re-adjustment at sentence and document level to provide a final re-improved language categories.

4.3.1 Document Loading

This module is devoted to load a document from its file path, selected by the user, and converting it into a string format. It is a fundamental module in order to successfully read the various words in the file. The Algorithm 5.1 illustrates the document loading process of this investigation.

Algorithm 5.1: Document loading

Input: - Document file path

Output: - Document text content in string format

Begin

 Select file path for loading

 Do

 Read the content of the file line by line

 Assign the content to string

 Until end file

End

As shown in Figure 4.2, the file is opened and its textual content is loaded line by line, building the resulting string.

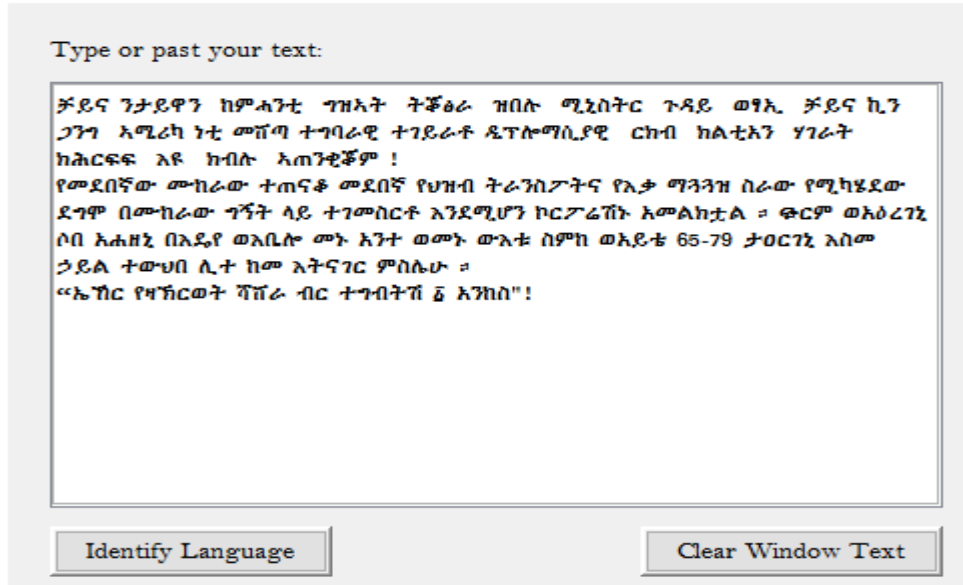


Figure 4.2: Text area of prototype after text document is loading

4.3.2 Sentence Segmentation

This module is a preliminary step and responsible to divide a string of input text document into meaningful units called sentences, before further processing. As shown in Figure 4.1 this module is fundamental particularly intended for testing phase, since identifying and indexing of sentence for the given text document is relevant for our sentence level reformation module. Sentence level reformation module; perform language classification adjustment of each words sentence by sentence, which needs an indexed sentence. In detail we will discuss about this issue under section of sentence level reformation. .

In order to achieve the process of dividing up a running text into sentences, an Ethiopian sematic language sentence boundary markers , which can be one from the characters ‘::’ , ‘?’ , ‘!’ , ‘;’ are used. As explained before, our language identifier is able to identify a language even if a text document with single word or phrase, meaningful units which are component of sentence and has not end of sentence markers. In this study such units are considered as a single sentence in order to make suitable our sentence indexing operation.

Algorithm 5.2 illustrates the sentence segmentation process of the present study. After the text loading is done, a string representation of text document feeds as input to this module and split by language end of sentence markers.

Algorithm 5.2: Sentence segmentation

Input: - Document text content in string format

Output: - Bag of indexed sentences

Begin

Count = 1

Sentence Separator = "?", ":", "!", ";";

If text contain one of Sentence Separator

 Split text into sentence by Sentence Separator

 If sentence is not empty

 Save count and sentence into bag of sentence

 Count = Count +1

Else

 Save count and sentence into bag of sentence

End If

End

During sentence segmentation, the sentence separator markers are not losing, since in this study the language of sentence separator markers also labelled to a language category contextually based on the rules developed for this investigation (in detail we will discuss under the rules section).

4.3.3 Tokenization

This module is responsible to represent a text document for purpose of language identification. It deals with the sentence segmentation into words, units which are meaningful for distinguishing between languages. It involves pre-processing of each sentence which is provided by previous sentence segmentation module, to deal with the issue of dividing each of sentences into words.

In this investigation the language identification is processed for each of individual words independently. This module is used for both training as well testing phase as explained in Figure 4.1, since in training phase as well as in testing phase in order to generate language profile and test document profile respectively , the tokenization module is mandatory.

A tokenization module is dedicated to tokenize the sentence into bag of words in order to making suitable for the next module to generate the target language profiles during training phase of the system. On the other hand, during testing phase in addition to tokenization of sentence, indexing or marking the position of each tokenized words within in a sentence is performed. This marking of word position in a sentence is used to identify the language switching points during language labeling of each individual words in a text document. So, finally this module provides a bag of words with sentence position as well as word position information are passed onto the next module for further processing.

In order to obtain the bag of words from the segmented bag of sentence, we substitute all occurrences of multiple white spaces with a single white space, and then split it by word separator i.e. white space. Algorithm 5.3 illustrates the process of generating bag of words. This bag of words contains sentence level as well as word level index information.

Algorithm 5.3: Tokenization for testing phase

Input: - Bag of indexed sentences, Document text content in string format

Output: - Bag of indexed words

Begin

For all sentences in bag of indexed sentences do

 Split each sentence with whitespace into bag of words

 For all bag of words do

 Retrieve start position of word from document text

 Put word with sentence and word position information

 End for

End for

End

Finally, we assign each single word with its sentence and word index to a Bag of words information that is provided to next module. As explained before, tokenization during training

phase involves only segmentation of the texts into words with whitespace and store into bag of words excluding word indexing information.

4.3.4 Indexing

As explained earlier, this module is responsible to provide the index information for both sentence as well as token of the given test document. As shown in algorithm 5.2, when the sentence segmentation is done the index assignment for each sentence in text document is assigned and this indexing information is very useful information in order to perform sentence level reformation, language category improvement at sentence level. The index is assigned for each sentence of text sequentially with sequential integer number i.e. 1...n and this index information is used as unique marker of each sentence of document.

On the other hand, the index information also provide to tokens of a test document after the tokenization module is done. However, this index information contains the actual position of each tokens in text and which is used later to identify the language switching point. The actual position of a word is used to identify the starting and ending position of tokens that labelled to a language category in given text document. As shown in above Algorithm 5.3, during tokenization module the actual start position of each token in text document is computed and recorded.

4.3.5 Normalization

As shown in Figure 4.1, this module is shared for both training as well as testing phase and it is concerned to normalize the homophone characters. In Geez writing system has homophone characters, characters which have same pronunciation but different symbols. For example, it is common that the character ስ and ሥ are used interchangeably as ስሬ and ሥሬ to mean “work”.

Consideration of these characters as different reduces our language identification performance, since during ngram computation a language profile as well as test profile it creates variation. Hence, those homophone characters should be considered in one manner or symbol for our language identification task. In this study, this normalization process is handled automatically by replacement of those homophone characters through representative character symbol. Table 4.1 shows the Ethiopian Semitic language homophone characters with corresponding replaced

characters. These homophone characters used in this investigation is adopted from a work done by [90].

Characters to be replaced	Replaced characters
ሐ,ሐ,ሐ,ሐ,ሐ,ሐ,ሐ	ሀ,ሀ,ሂ,ሃ,ሄ,ሀ,ሀ
ኀ,ኀ,ኀ,ኀ,ኀ,ኀ,ኀ	ሀ,ሀ,ሂ,ሃ,ሄ,ሀ,ሀ
ዐ,ዐ,ዐ,ዐ,ዐ,ዐ,ዐ	አ,አ,አ,አ,አ,አ,አ
ሠ,ሠ,ሠ,ሠ,ሠ,ሠ,ሠ	ሰ,ሰ,ሰ,ሰ,ሰ,ሰ,ሰ
ከ	ኮ
ኀ	ኀ
ወ	ወ

Table 4.1: Normalized Characters

Algorithm 5.4 illustrates the processes of representing the homophone characters with normalized character symbols in order to reduce the incorrect feature extraction of our language identification task.

Algorithm 5.4: Normalization for training and testing phase

Input: - Bag of indexed words

Output: - Bag of indexed normalized words

Begin

For all bag of indexed words do

 Lookup a word characters along with homophone character list

 If a word contains a homophone character then

 Replace a word characters with corresponding normalized value

 Else

 Return a word as a normalized word

 End if

End for

End

In addition to this, particularly for training phase this module is also responsible to clean or remove all unnecessary characters (i.e. all special characters, digits) to build language profiles. Since, these special characters are not a member of a particular language, rather a member of all Ethiopian sematic languages. Hence, these characters are not involved on build a language profile of a particular language. However, nothing is removed during testing phase, since all parts of text document either special characters, digits or other words are expected to be label with one of the language category.

4.3.6 Character Ngram Extraction

As stated in chapter 2, character ngram is a popular approach used for different NLP classification tasks. A character ngram is a set of n consecutive characters extracted from a word and the main motivation behind this approach is that words in the same language will have a high proportion of character ngrams in common.

As showed in the Figure 4.1, this module is shared by both training and testing phase. In this investigation, this technique is used in a different way for general purpose language identification, a capability of identify a language for both monolingual and multilingual textual documents in once. In addition to this, the text document which is given to be identified can be at any level, it can be a document which contains a single word, phrase, sentence or collection of sentences or paragraphs. Moreover, it is robust to grammatical errors, for example the word “እንዳይከሰት” and “እንዳይከሰ” share the majority of character ngrams.

This module used sequence of characters as features and concerned to build a language profile as well as a target test document profile. In this study in order to enhance our discriminating performance between languages, we include a relevant feature of sequence of ngram characters extracted from a given word. These features are character ngram occurrence and character ngram location occurrence and the result of both feature sets is compared, we would see the result under experiment section of this study.

One of the contributions of this work is consideration of ngram location in a word to classify any types of documents (i.e. monolingual or multilingual documents). A language like Ethiopian sematic language families are difficult to identify each other, since they are very similar and use the same script. So, in order to enhance our capability of language discrimination, we include ngram location feature set in addition to character ngram. Since we observe that in Ethiopian sematic languages used in this investigation has differ to each other based on occurrence of ngram location in a word, in which a particular characters and sequences tendencies to occur at the particular location pattern.

Furthermore, in order to capture the entire word for these short words, we pad each word with (n-1) special characters to denote the beginning and end of a word, and use n-grams extracted from these modified words. For example, in a 3-gram setting, from the Amharic word ፍ, we derive the 3-grams \$\$\$, \$ፍ#, and ፍ##, with \$ፍ# indicating that the entire word is represented.

In order to develop a multilingual language identifier, it needs a multilingual training corpus to train a language identifier. Some researchers are use such labeled multilingual training corpus for resourced language to develop a multilingual language identifier, however for under resourced language like Ethiopian Semitic language there is no any such labeled multilingual corpus. So, it is difficult to develop multilingual language identifier which is trained form such labeled multilingual training data. In order to alleviate such difficulty, in this investigation a monolingual row text of a language is used to train our general purpose language identifier. So, our general purpose language identifier is train from a monolingual row text of language and able to classify either monolingual or multilingual text document. In addition, it is easy to extend our general purpose language identifier to support other languages, since it only needs a training monolingual row text of a language. Moreover, in order to observe the capability of the proposed general propose language identifier we used a character ngram model in two different forms: fixed character ngram size and infinity ngram size.

4.3.6.1 Fixed Length Character ngram

It is a familiar text representation technique, in which a text document is represented with a bag of character ngrams with a selected optimal size of N . In this investigation; we explore the optimal maximal length ngrams empirically; from 2 up to length 5, see the results in chapter six.

As explained previously, this module is used for both training and testing phase. In case of training phase the language profile with ngram character (i.e. $n = 2 \dots 5$) is build and similarly in case of testing phase each extracted word from a test document is also represented with sequence of character ngram in order to achieve our general purpose language identifier.

For example an Amharic word “**እንዳይከሰት**”, which is extracted from given testing document can be represented with the following character ngram with fixed size $N = 3$ called trigram.

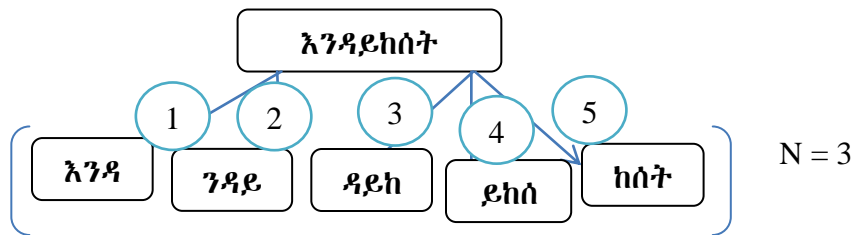


Figure 4.3: Character trigram representation of word “**እንዳይከሰት**”

As shown in above Figure 4.3, after previous module is processed each bag of normalized words of a text document is representing in sequence of characters depending on the size of ngram and checked along each language profiles in order to determine the language category of a particular word.

As shown from the above example, the word “**እንዳይከሰት**” is represented with 5 different character ngram features using character ngram size of 3 called trigram. In order to compute the character ngram probability of the given word “**እንዳይከሰት**” based on the extracted character ngram features, the ngram probability of each extracted character ngrams are taken from each of the domain language profile information. Since, each character ngram type extracted from a bag of word is stored in a language profile with their relative frequency or ngram probability value as language profile model.

As describe in Algorithm 5.5 each word from bag of indexed normalized words is feed as input to character ngram extractor module and before extraction of character ngram for a given word, padding of N-1 start and end character size is done in order to capture the entire characters of short words. After such character padding, the extraction of a word with padded characters is done for all ngram size $N = 2$ to 5. In order to extract the character sequence from a word it depending on the ngram size with until N-1 character index. Hence, a loop is used to navigate these extracted character sequences of a given word and add to bag of character ngrams.

Algorithm 5.5: Character Ngram extraction for both training and testing phase

Input: - Bag of indexed normalized words

Output: - Bag of character ngrams

Begin

For $N = 2$ to 5 do

 For word in bag of indexed normalized words do

 Padding a word with N-1 start and end character size

$j = N - 1;$

 For $i = 0$ to $len - j$ do

 Ngram Type += word character at index i to j

 Add Ngram Type to Bag of character ngrams

 End for

 End for

End for

End

The Algorithm 5.5 is used to extract the sequence of character ngrams from bag of words in order to build a language profile during training phase and to build a testing profile during testing phase.

4.3.6.2 Infinity ngram

The approach described above is able to prove fixed length character ngrams and as shown in above example of Amharic word “አንዳይከሰት”, an extracted features of ngram types with trigram ($N = 3$) is too small and it is difficult to build a powerful tool which can detect a language at word level. Since as an extracted ngram features for a given word are more in number or rich, and then a reliability of language detection for a given word is increased. So, in order to make the proposed general purpose language identifier more powerful for any input document, an approach which can able to extract rich or more ngram features at word level required.

To achieve this we used a new approach which can able to extract a combination of all ngram size features of a word in once and we call this as Infinity ngram. This approach have been introduced by [89] for document classification, which extracts all character ngrams of a string as features for document classification task. The researchers [89] use this new approach, since tokenized words are not enough for determining a class of a document, ultimately through experiment learning a classifier by using all character ngram features set achieves a better result.

Hence, in order to get such benefit of Infinity ngram, we adopt this new approach for word level general purpose language identification task. During Infinity ngram the size of ngram used to extract character sequence for both training and testing is not fixed rather it depends on the given word length. The size of ngram in Infinity ngram vary from word to word and it depends on a word length, maximum at $n = |w|$. So, all ngram types range from 2 to $|w|$ is extracted to represent a given word with length w maximum of ngram. This novel approach produces a considerably large character ngram feature set when compared with fixed length character ngram representation.

As shown below in Figure 4.4, from previous example for Amharic word “አንዳይከሰት” it is possible to extract 28 numbers of character ngram features. This is very large numbers of ngram features when compared with fixed length character ngram approach and this enhances the proposed general purpose language identification task at word level.

The number of extracted character ngrams using this approach is very large and this is very rich feature set to make language labelling at word level. Hence, infinity ngram is a better approach for language labelling at word level.

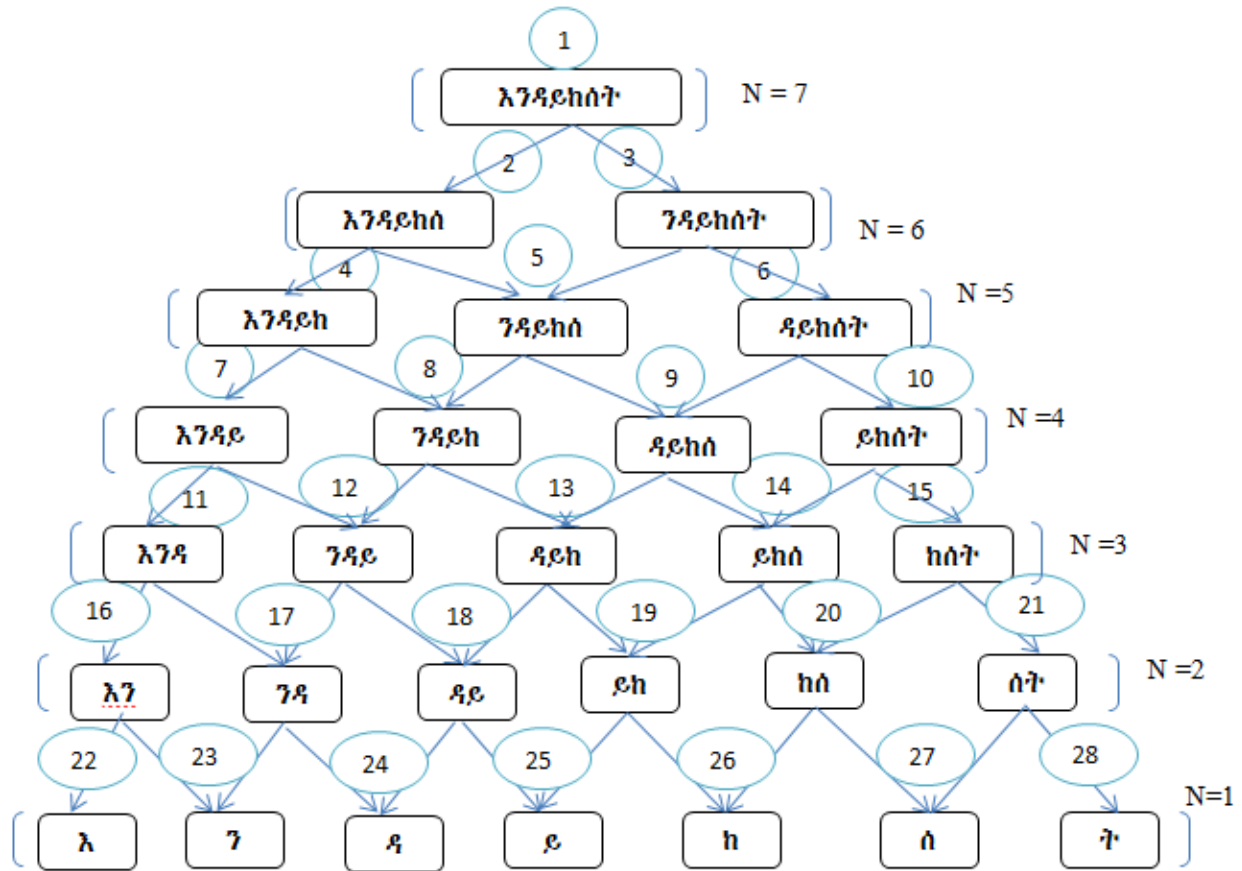


Figure 4.4: Infinity ngram character representation of word “አገዳይከሰት”

In this investigation character ngram with size of $N \geq 2$ is taken to represent a given word, since for $N = 1$ all Semitic languages are similar pattern, not useful ngram type for our word level language identification task. As explained earlier, the size of N may vary from word to word, it depends on the string length of a given word, and means that it is possible extract Ngram $N = 2 \dots |w|$ as shown in above Figure 4.4. So, this new approach is used in this investigation in order to extract all character ngram types for a given word to design and develop an outperform word level based language identifier.

On the other hand, we experimented with a wide range of values to weight the ngrams extracted from the infinity ngram approach, since as showed in above example Figure 4.4, as N (ngram

size) increase, the more capability to express the given word. Hence, a weighting factor is taken for each character ngrams with N-1. Surprisingly this weighting factory enhances the performance of the word level language identifier based on infinity ngram approach. We used N-1 as a weighting factor and we name this as ngram weight factor, W_{ngram} . This weighting factory involves on the decision of the language label of a word. Finally, the probability of each ngram extracted by infinity ngram depending on the ngram size N can be computed as

$$\text{Prob}_{\text{ngram}} * (N-1) \quad (5.1)$$

Where $\text{Prob}_{\text{ngram}}$ is an actual ngram probability taken from target language profile and N is ngram size used to extract ngram features from a given word. As mentioned above this weighting value vary depending the size of ngram N and as N becomes large the ngram weight increase. This weighting factor enhances the weighted probability of a given ngram, means that the ngram has more expressive power on predicting a language category of a given word.

From previous example for Amharic word “አንዳይከሰት”, to compute the ngram probability of a given word all ngram types having ngram size $N \geq 2$ are taken. As explained before, the words “አንዳይከሰ” , “ንዳይከሰት” with ngram size $N = 6$ is more expressive the actual word “አንዳይከሰት” than the ngram types “አንዳይ”, “ንዳይከ”, “ዳይከሰ”, “ይከሰት” with ngram size $N = 4$. Hence, the language profile having best probability with ngram type with $N = 6$ is more probable to be language category of given word “አንዳይከሰት” than $N = 4$ ngram types. To maintain this truth during ngram probability computation of all character ngram types, ngram weighting factor is used depending on size of N. For more clarification see the following ngram computation depending on the previous example:-

For $N = 6$

$$\text{Prob}_{\text{ngram}} * (N-1) \Rightarrow \text{Prob}_{\text{ngram}} * 5$$

For $N = 4$

$$\text{Prob}_{\text{ngram}} * (N-1) \Rightarrow \text{Prob}_{\text{ngram}} * 3$$

During infinity ngram approach, the language profile as well as words extracted from tested document is represented with all ngrams depending on the word length. So, in order to speed up our searching process of ngram types along each of language profiles we used ngram type length as index. The searching process is the most important and potentially the most time consuming activity in the whole process, since all ngram type extracted from a given word with different N size need to be checked against all ngrams in the language profile. In order to enhance the searching process, we used length of ngram types as index. The idea is that ngram type extracted from a given word having $N = 3$ need to be checked in only with those ngram $N = 3$ in a language profiles. This makes the searching process of proposed language identifier based on infinity ngram approach faster, since there is no point in search for ngram size $N = 3$ among other ngram sizes where it will never be found.

Thus, to speed up the process by organizing the character ngram language profile information by its ngram length as index and this would be a one-time process such that once the language profile is indexed in this way it is only updated as and when necessary.

As explained in Algorithm 5.6, given a word as input form bag of indexed normalized words provided by previous process and to generate all non-empty character ngrams with any ngram sizes for a given word, we used three nested loops. The outer most loops picks starting characters; middle loop considers all characters as ending character of ngram type. The inner most loops puts characters from currently picked starting point to picked ending point. In order to include those ngrams which are more discriminative for a given word, we exclude ngram size of 1 ($N = 1$), since it is less discriminative feature for our word level language identification task.

Algorithm 5.6: Infinity ngram extraction for both training and testing phase

Input: - Bag of indexed normalized words

Output: - Bag of character infinity ngrams

Begin

For all bag of indexed normalized words do

STR = Character array of a word

For len = 1 to length of STR do

For i = 0 to length of STR – len do

j = i + len -1

For k = i to j do

If len > 1 then

Put STR[k]

End if

End for

End for

End for

End for

End

In order to achieve the proposed general purpose word level language identification task, for both fixed length character ngram as well as infinity ngram, the relative frequency or probability of each ngram is computed. The ngram probability of each n-gram X_i in a language L_j is computed by a formula in Equation 5.2.

$$Prob(X_i^j) = \frac{f(X_i^j)}{\sum_{i=1}^n f(X_i^j)} \quad (5.2)$$

Where, $f(X_i^j)$ is the frequency of ngram X_i in the language L_j and $\sum_{i=1}^n f(X_i^j)$ is the total sum of character ngram type occurrence in language L_j .

During language profile construction with either fixed length character ngram or infinity ngram the number of times each ngram occurs in the training corpus of each language is computed. It outputs the relative frequency or weight of each unique ngram using a formula in equation 5.2.

On the other hand , the formula in equation 5.2 also used for both character ngram approaches (i.e. fixed character ngram and infinity ngram) with ngram location feature set to compute ngram location relative frequency or location based ngram probability of each ngram type.

Beside this, in this investigation during language labelling decision of current word contextual information is used, since during language labeling of a word the language category of a contextual word matters. To acquire this contextual information, during the ngram probability computation of the current word the ngram probability of a previous word in each of the target language in a set of domain language is used with the formula in equation 5.3. Since, during language category decision of a current word the previous word language category is considered. Hence, in order to keep this information a merged ngram probability of current word and previous word during language label decision of current word is used and we call this merged ngram probability as augmented probability. This augmented probability is used in both fixed length character ngram and infinity ngram with or without location feature set.

$$AgmProb (X_i^j) = \frac{Prob(X_{i-1}^j) + Prob(X_i^j)}{2} \quad (5.3)$$

Where $AgmProb (X_i^j)$ is improved probability of a current word X_i for a set of target language L_j , $Prob(X_{i-1}^j)$ is ngram probability of a previous word in language L_j and $Prob(X_i^j)$ refers to the ngram probability of current word X_i in language L_j .

We can observe that, the first word of test document cannot have a previous word and during this case there is no an improvement of ngram probability of a word with contextual information as we explained in equation (5.3). Hence, in such case the ngram probability of a word is not augmented, rather a pure ngram probability of a word is taken.

4.3.7 Word ngram

Word ngram is a widely used feature type in many text classification tasks including text language identification. In this study a word ngram with size of 1 ($n=1$) is adopted to represent the language profiles of all supported languages with all unique words of the language. Such word ngram language profile information is used to compute an exact matching for the incoming word. However, in this investigation word ngram approach not devoted independently for language identification, rather it works in combination with character ngram.

Therefore, in this investigation the language labeling decision for each individual word extracted from test document is not only based on the ngram language probabilities; the process also takes the language label of the previous word into account to consider contextual information. If the current word is in the lexicon of the language of the previous word, the current word is tagged as that language, without considering the current word's language probabilities. For words in the lexicons, this simple decision was found to be more effective than character ngram computation based language identification. If the current word is not in the lexicon of the previous word, we consider the language identity probabilities rather than the label of the previous word;

The process of this approach starts with the construction of the language models, which are generated by tokenizing the training sets in the various languages and eliminating duplicate words after pre-processing. The resulting language models are word lists comprising unique occurrences of words in each language. The system then computes by searching for each given word along the training language profiles.

For the first word of test document, there will be no previous word and the language labelling decision is with exclusion of word ngram approach. Hence, the language identification of the first word of a document is always done either with character ngram probabilities or rules. In general, the word ngram approach is functional when a current word has at least one previous word, since this approach always needs contextual information of previous word.

Algorithm 5.7 illustrates how the word ngram approach searches a given word along the lexicon of the previous word language category in order to check that a current word is a member of the previous word language category or not.

As explained in algorithm 5.7, like that of character ngram , to check the existence of current word along the lexicon of previous word language category a word length is used as search index. This is used to enhance the searching capability of a word within previous language category. When a word exists in a previous word language lexicon, an algorithm returns true, unless return false to the classification module of the proposed approach.

Algorithm 5.7: Word ngram checker

Input: - Bag of indexed normalized words

Output: - language category

Begin

For current word in bag of normalized words do

If current word not first word of test document then

For each lexicon words in L (language of previous word) do

 If len equal to index of lexicon word then

 If current word match with lexicon word then

 Return true

 Else

 Return false

 End if

End for

End if

End for

End

Use of most frequent word dictionaries is a popular method in a language identification task, which keeps only high frequency words, but that is suitable for longer texts, and surely not for language mixing situations. The proposed language detection approach is targeted at the word level and for short texts, so we cannot only rely on the most frequent word lists and have thus instead used the full length dictionaries extracted from the training corpus.

4.3.8 Rules

In this study there is also an investigation of using heuristics or rules which are used in combination with the word ngram and character ngram approach for better performance of the proposed general purpose language identification task. The rule based method utilizing manually constructed rules developed by language experts.

As explained earlier, the special characters and numerical characters (i.e. both Latin and Geeze numerical values) are treated in a different way during testing phase of the proposed approach. In training phase these characters are removed during the preprocessing steps, since they are not particularly discriminate a particular language rather used commonly in all used Ethiopian sematic languages. Therefore, such features are not useful to build a language profile and they are excluded. However, during testing phase they are not removed from testing document, since the user may confuse about elimination of these words.

Consequently, in this investigation in order to overcome the language labeling of such characters and other words that contains unique characters of a particular language the following rules are developed.

(I). Rules for words contains unique characters of particular language

- a. If a word contains Guragigna unique characters which are stated in Table 4.2, then a word language category would be label as Guragigna.

ሙ , ማ , ሚ , ም	ገ, ገ, ገ, ገ, ገ, ገ, ገ
ቡ , ቤ, ቦ	ኀ, ኀ, ኀ, ኀ, ኀ, ኀ, ኀ
ሪ, ሪ, ራ, ሬ	ቀ, ቀ, ቀ, ቀ, ቀ, ቀ, ቀ
ፑ, ፑ, ፑ, ፑ	ቨ , ቨ , ቨ, ቨ, ቨ, ቨ, ቨ
ኸ, ኸ, ኸ, ኸ, ኸ, ኸ, ኸ	ኀ, ኀ, ኀ, ኀ, ኀ, ኀ, ኀ
ሸ, ሸ, ሸ, ሸ, ሸ, ሸ, ሸ	

Table 4.2.: Guragigna specific characters

For instance, a given word “ጭርም” labeled as Guragigna language category, since it contains ‘ጭ’, is one of character that indicates a word is Guragigna. So, like this if a word contains one of the characters listed in above table 4.2, it would be labeled as Guragigna language category.

- b. If a language category is ambiguous for a single word and a word contains one of characters mentioned in table 4.3, when the language category ambiguity is occurred between Amharic or Geeze with Tigrigna or Guragigna, then the language category is labelled as Tigrigna or Guragigna.

ቀ, ቀ, ቀ ,ቀ, ቀ ,ቀ, ቀ	ቈ ,ቀ, ቀ ,ቀ ,ቀ
ኸ, ኸ ,ኸ	

Table 4.3.: Tigrigna and Guragigna unique characters

For instance, a given word “ጉቆፊ” after checked up with all components of the investigation approach and when a word labeled as Amharic and Tigrigna language category. So, according to this rule it would be labeled as Tigrigna language category, since a word “ጉቆፊ” contains a character ‘ቆ’, is one of a character that indicates a language is either Tigrigna or Guragigna language category.

(II). Rules for words with digit

- a. If the first character of word is 'ብ' or 'ገ' followed by any digit, then it would be assigned as Tigrigna languages category. For instance, a given word ብ2007 would be labelled as Tigrigna language category.
- b. If a word begins with any digit and ends with 'ይ' or 'ይን' or 'ታት', then it would be assigned as Tigrigna language category. For instance, a given word 1999ታት would be labelled as Tigrigna language category.
- c. If the first character of words is 'በ' or 'ለ' or 'የ' or 'ከ' and followed by any digit, then it would be assigned as Amharic languages category. For instance, a given word በ1983 would be labelled as Amharic language category.
- d. If a word begins with any digit and ends with 'ኛ' or 'ቱ' or 'ሺ' or 'ኛው' or 'መቶ' or 'ዎቹ' then it would be assigned as Amharic language category. For instance, a given word 8ኛ would be labelled as Amharic language category.

(III). Rules for pure Latin digits and geez numbers

If a string is pure digit or geez numbers, then the previous word language category would be assigned as language category of a string.

(IV). Rules for punctuation marks

If a string is punctuation mark then the previous word language category would be assigned as language category of a string.

In addition to this, even if rare but there is a probability of occurrence of a language category having equal ngram probability computation. In this situation the proposed approach has tried to disambiguate using rule stated in under rules for words contains unique characters of particular language subsection of rule b. If this rule is not able to disambiguate such station, the proposed approach tries to disambiguate a language category using word contextual information. When lists of language categories has equal ngram probability value after all computation and from the list of language categories if a previous word language category is found, it assigned as a

language category of a current word, unless one of the language from the least will be assigned randomly. Even if the language category is assigned with final option randomly from the list of ambiguous language category, it would be adjusted with the proposed approach sentence level module and document level reformation modules.

4.3.9 Classification

This module is concerned to correctly guess the language $l \in L$ in which each word of a text document is written, when all languages in the set L are known to the language identifier. To classify each word of an input text document with regard to the language models, the distance between them are calculated. The language with the minimal distance to the word of a text document is chosen as the language of the given word.

Many of the learning algorithms applied to both monolingual as well as multilingual language identification in the framework of Bayesian classification and achieved better performance. So, in this study a Bayesian classifier is adopted to achieve the proposed general purpose language identifier.

A naïve Bayes classifier uses the concept of Bayes' theorem [83]. This classifier assigns the most likely classes to an input string, based on the highest a posteriori probability, given the input string.

For text language identification purpose, a naïve Bayes classifier can be constructed using ngrams as features. Let T be a set of training samples and let each sample be represented by n feature vectors, $X = x_1, x_2 \dots x_n$, with their class labels. Let there be n classes: $L_1, L_2 \dots L_m$. to predict, a sample X_n is selected to belong to class L_i , if and only if:

$$P(L_i/X) > P(L_j/X); \text{ for } 1 \leq j \leq n; j \neq i \quad (5.4)$$

Where $P(L_i/X)$ is the probability of a class L_i given a sample X. Bayes' theorem states that:

$$P(L_i/X) = \frac{P(X/L_i)P(L_i)}{P(X)} \quad (5.5)$$

Where $P(L_i/X)$ represents the likelihood of a sample X belonging to class L_i , and $P(X)$ does not influence model comparison.

The class a priori probability $P(L_i)$ represents the count relative frequency in the language profiles, so that $P(L_i)$ can be omitted as well. According to the Naive Bayes assumption, statistical independence of features is assumed, and the class L_i is selected such that $\prod P(x_j / L_i)P(L_i)$ is optimized, where $P(x_j / L_i)$ is then the likelihood of a specific ngram being observed in a given language profile, and the word being classified consists of j n-grams.

Furthermore, the classification module is not only categorize based on the character ngram approach, but also accepts values return from previous word ngram module. The word ngram is a Boolean module which returns true if a word is found in the lexicon of previous word language category and the classification module assigns the previous word language category as language of current word, unless no assignment.

On the other hand, during Bayes based ngram probability computation there is rare or unseen n-grams can result in poor probability estimates. In order to eliminate this problem of poor probability estimation a smoothing technique is used in this investigation. ‘Smoothing’ refers to a range of techniques that re-distribute probability density among rare or unseen tokens [84]. As maximum likelihood estimates (MLE) are used by Naive Bayes classifiers to estimate class probability, smoothing can help to address poor probability estimates, which result in zero probability of missing n-gram sequence models.

In T-LID task [84], different smoothing techniques have been proposed and applied. Such as: additive smoothing, Katz smoothing [85], Witten-Bell smoothing [86], absolute discounting [87], Kneser-Ney discounting [87] and Jelinek-Mercer [88] methods. This study adopts additive smoothing from the above mentioned smoothing techniques, because of its simplicity of implementation and suitable for the proposed general purpose language identification task.

4.3.9.1 Additive Smoothing

One of the first to experiment with character n-gram models in language identification was Dunning (1994). The researcher used character-based language models smoothed with the simple “add one” smoothing originating from Laplace’s rule of succession, which adds one to the counts of all possible n-grams. A simple generalization is the general additive (Lidstone) smoothing, where a smaller value $\lambda < 1$ is added to the counts:

$$P_{add}(X_i | X_{i-n+1}^{i-1}) = \frac{c(X_{i-n+1}^i) + \lambda}{c(X_{i-n+1}^{i-1}) + \lambda V'} \quad (5.6)$$

Where x_j denotes the sequence $x_i \dots x_j$, V is the size of the vocabulary (number of different characters in the language) and $C(x)$ denotes the number of occurrences of an item x . The parameter can be optimized by maximizing the probability of the held out data. In this study, an additive smoothing with $\lambda = 1$ is called Laplace smoothing or add one smoothing is used.

4.3.10 Sentence Level Reformation

As shown in Figure 4.1, after the classification module each word of a test document is assigned to one of a language category in set of domain language and we called this classification result as pre-classified language categories. However, due to very similarity of Ethiopian Semitic languages there is a station of wrong language category assignment to a given word. Hence, in order to adjust such incorrect language category labeling we include a module called sentence level reformation.

The sentence level language reformation is given by averaging the word level language prediction results at a sentence level. The output of language category result from previous classification module is transformed into dominance language category at a sentence level if and only if the average occurrence of a particular language is equal or above the defined threshold value, a value which is selected through the experiment. The average occurrence of each set of language labelled in a given sentence is computed with formula in equation 5.7.

$$avg(L_i^j) = \frac{f(L_i^j)}{\sum_{i=1}^n f(L_i^j)} \quad (5.7)$$

Where, $avg(L_i^j)$ is the occurrence of language L_i in the sentence j and $\sum_{i=1}^n f(L_i^j)$ is the total sum of language occurrence in the sentence j .

This module is concerned to compute language dominance at sentence level from pre-classified language categories result. When a dominance of a particular language satisfies a specified language dominance threshold value then the language categories of each word within a sentence is re-improved to dominant language category. Algorithm 5.8 illustrates the sentence level reformation process of the present study. After the language classification result is provided by classification module, the sentence level reclassification is done for each language category result of an indexed sentence of a text document.

Algorithm 5.8 depicts that The input of this algorithm is pre-classified language categories provided by previous module and for all words language category in each indexed sentence is computed their language label occurrence. After computing the total occurrence of each language category of a word in a sentence, compare the total occurrence of each language category and choose a language category with maximum occurrence value.

When a the number of chosen language category within a sentence is more than one , hence sentence level reformation is not required , unless the average occurrence of chosen language category is computed through dividing total occurrence of chosen language by the sum of total occurrence of all languages in a sentence. Finally, when the average occurrence satisfies the given threshold value then the initial language category result is adjusted by dominant language category.

Algorithm 5.8: Sentence level reformation

Input: - pre-classified language categories

Output: - sentence level reformed language categories

Begin

For all indexed sentences in test document do

 For all assigned language categories in sentence do

 Compute total occurrence of each language category

 If total occurrence > 0 Then

 Compare total occurrence of each language category and choose language with max occurrence

 If number of chosen language category > 1 then

 Continue

 Else

 Average_occurrence = total occurrence / sum of total occurrence of all language in sentence

 If Average_occurrence >= thresholdValue then

 Change previous assigned language category of each word in a sentence to dominance language category

 Else

 Continue

 End if

 End if

End for

End for

End

4.3.11 Document Level Reformation

This module capable the proposed language identifier to have best performance for language identification of monolingual documents (documents written in one language). After the sentence level reformation is done, this module is dedicated to compute the language reformation at a document level.

Document level reformation is the process of adding improvement on a language category result reformed by previous sentence level reformation module through making adjustment at a document level as whole. Since, the given test document may be monolingual document and this module helps to adjust incorrect language labeling of monolingual documents into more than one language category.

When a dominance of a particular language occurrence satisfied document level language dominance threshold value then the language category of each word with in text document as whole is reformed to a single dominant language category. The document level threshold value used for document level adjustment is defined based on the investigation experiment result.

The average occurrence of each set of language labelled in a given document is computed with formula in equation 5.8.

$$avg(L_i^j) = \frac{f(L_i^j)}{\sum_{i=1}^n f(L_i^j)} \quad (5.8)$$

Where, $avg(L_i^j)$ is the occurrence of language L_i in the document j and $\sum_{i=1}^n f(L_i^j)$ is the total sum of language occurrence in the document j .

In a similar fashion the algorithm 5.8, illustrated above is adopted to our document level reformation module. However, in this module the reformation is not at a sentence level, rather it is at a document level as a whole.

4.4 Prototype

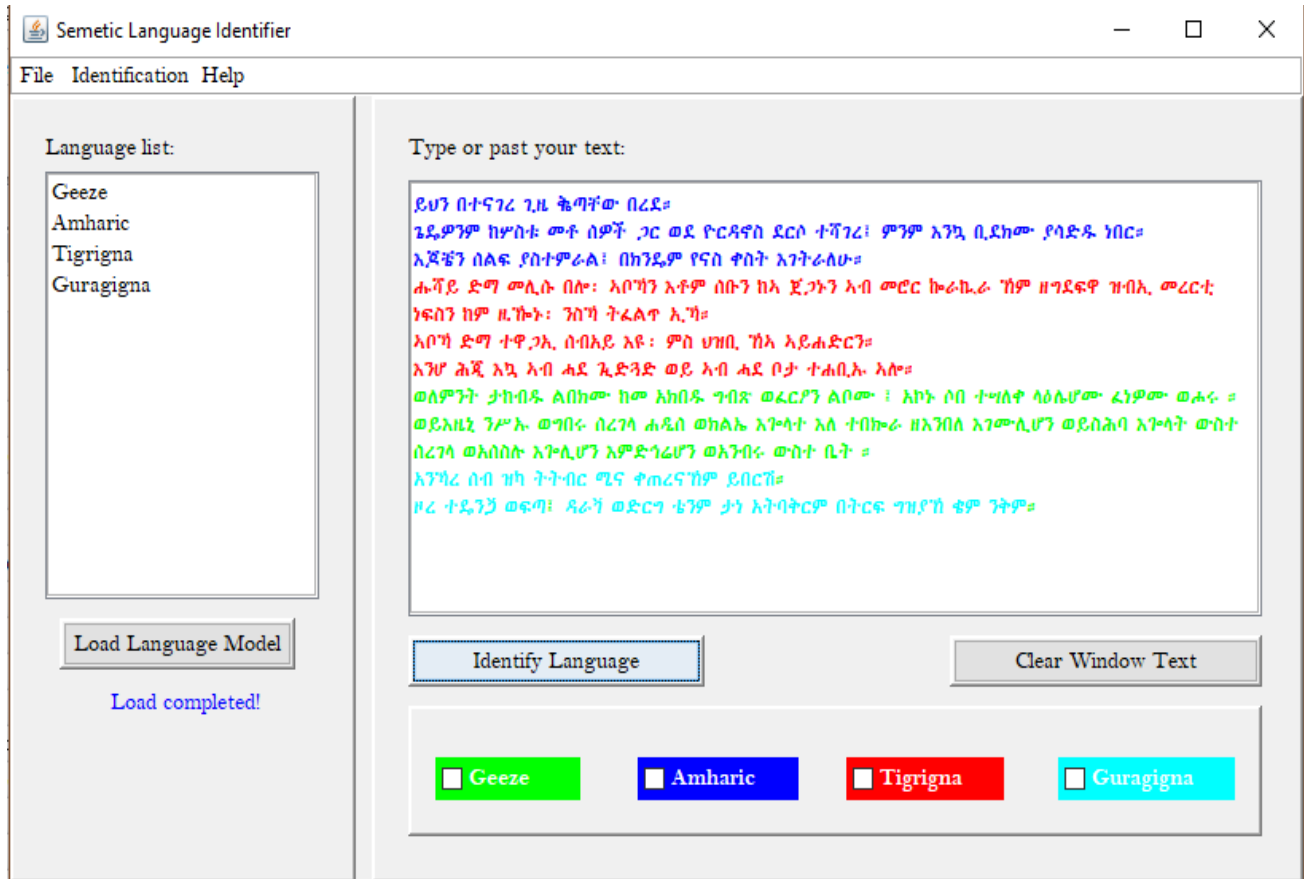


Figure 4.5 the Proposed General Purpose Language Identifier Prototype User Interface

Figure 4.5 illustrates the prototype of General purpose language identifier user interface and which contains three sections.

- Loading language profile section:** - A section in which each language profile model is loaded when “load language model” button located below this section is click by the user. The completion of loading the language profile model is indicated when the “load completed!” message is displayed as shown in Figure 4.5.

- **Loading text document section:** - a section in which the content of test document is loaded when the user select the file with file selection menu located on the top of prototype interface. As explained in Figure 4.2, the text of test document is black color before the language marking is done.
- **Language marking section:** - as shown in figure 5.5, this section presents the language marking part of the proposed general purpose language identifier. The indication of each language label is represented with four different colors: Geeze as green, Amharic as blue, and Tigrigna as red and for Guragigna cyan. When the user click the “Identify Language” button, as shown in Figure 5.5, the original black color of loaded text of test document is marking with previously color representation of each language category.

4.5 Summary

In this chapter, we briefly discussed the basic design goal and a proposed architecture of the proposed general purpose language identifier; it can identify the language of input test document at any level i.e. at word, phrase, and sentence as well as document level. Beside this, it can able to identify the language of monolingual or multilingual.

In this chapter, each proses of the proposed architecture of general purpose language identifier, steps in both training as well as testing phase have been briefly described. The design and implementation issues of these proses of training as well as testing phase of proposed architecture is presented briefly. Finally, we have also discussed about the prototype interface of this investigation briefly.

Chapter Five

Experiment

5.1 Introduction

As an evaluation measure, we apply accuracy, which is the proportion of correctly identified test samples. For individual languages, we can calculate precision and recall. Precision is the proportion of correctly classified test samples in all samples classified to the given language, whereas recall is the proportion of correctly classified test samples in all samples of the given language. Finally, experiment is conducted on proposed general purpose language identifier approach to measure its capability and performance. In order to achieve this, the corpus is dividing into 90% of training set for training purpose and 10% of testing set for testing purpose using tenfold cross validation technique. In this chapter, the detail experiments conducted for this thesis work are described briefly.

5.2 Data collection

Our method is a Rule based and machine learning process that derives statistics from a training corpus. We first train and test the method using the corpus.

In the literatures for text-based language identification ranges of methods have been used to compute classification accuracies. Classifiers were trained with different amounts of textual Training data; in some, documents were limited to one domain, and others would span over a Few domains. The various ways of measuring the size of the test strings includes number of characters, number of words, size in bytes which will depend on the encoding scheme used, lines, and sentences.

Some tests were performed on languages without any family relationships and others within language families. In studies where classifiers were compared against each other more reliable conclusions can be made, since the classifiers were evaluated under the same conditions. But comparison of results between different studies is usually difficult. In this study languages used for the experiment are from the same linguistic family; all are languages in Semitic family.

In order to develop a LangID system that is accurate regardless of characteristics or peculiarities of text from a particular source, we make use of data from a various sources. This will allow us to identify the characteristics of each language that are indicative of the language independently of the source that the data is drawn from.

By maximizing the variation between the sources, we maximize our ability to identify the general characteristics of languages that we can exploit to achieve multilingual LangID. Maximizing variation in our data sources is also critical for evaluating a LangID system, as we need to show that the system is robust across the types of variation found in our data sources.

For the purposes of this thesis, we assume that the document is represented in text form in some machine readable and human interpretable encoding, though we do not assume that the actual character encoding (e.g. ASCII or UTF8) is known in advance. In other words, in this thesis we only deal with LangID for digital text.

We explicitly exclude audio documents and images of text documents from consideration. Furthermore, we will focus on multilingual documents (documents containing text in more than one language) and monolingual documents, i.e. documents that we assume to contain text from only one language.

5.2.1 Data Sources

Machine learning approaches require datasets that maximize variation in individual languages, so that we may focus on determining characteristics of each language that are independent of the source-specific variation. In the rest of this chapter, we describe a number of data sources and the sources of variation that they capture. We also describe the dataset that we construct from each source for the purpose of this thesis.

5.2.2 Bible

The Bible is a collection of religious texts that is considered sacred in a number of interrelated faiths. In historical terms, the document has a complex history, and different denominations make use of different subsets of the texts as their own canonical version. Even where there is

agreement between groups on which texts are sacred, there can be subtle variations in the particular translations used. Nonetheless, Bible-derived corpora are attractive for LangID research because they typically provide a reasonable amount of well-curated text. Translations are often prepared and maintained by religious organizations around the world as part of missionary efforts. A number of previous authors have made use of text from Bible translations in LangID research (Hammarström 2007; Vatanen et al. 2010; Chew et al. 2011; Brown 2013).

In this thesis, we use a Bible Corpus assembled by Content bible.org for Amharic bible and geezexperience.com. Content Bible societies of Eritrea for Tigrigna bible have both Old and New Testaments but for Geez it has only some parts of the bible.

5.2.3 Books

There are different types of books which are written for different purpose like politics, bibliography, history, culture, poems etc. Those parts of the book hold different domain and the following tables show lists of books which are used in this thesis. For Amharic, Tigrigna, Geez and Guragigna see tables 5.1, 5.2, 5.3 and 5.4 respectively.

NO	Title	Author	Publication year	Place of publication
1.	አጭር የኢትዮጵያ ታሪክ ከንግሥተ ሳባ እስከ ዳግማዊ ምኒልክ	ምርመራና ከሥርዓት ትምህርት ዲሬክሰሊዮን	፲፱፻፶፩ እ.ኤ.አ.	ትምህርትና ሥነ ጥበብ ሚኒስቴር
2.	አግዐዚ	ብላታ ወልደ ጊዮርጊስ ወልደ ዮሐንስ	፲፱፻፷፩ ዓ.ም	ቅዱስ ጊዮርጊስ ማተሚያ ቤት
3.	አልወለድም	አቤ ጉበኛ		
4.	አሉላ አባ ነጋ	ማሞ ውድነህ		ኩራዝ አሳታሚ ድርጅት
5.	ዳግማዊ አጤ ምኒልክ	ጳውሎስ ኞኞ	በየካቲት ወር 1984 ዓ.ም	ማንኩላ አሳታሚ
6.	ዳግማዊ ሚኒልክ	አፈወርቅ ገ/አ.የሱስ	በሽህተ፱፻፩ ዓ.ም	ሮማ ከተማ
7.	የእቴጌ ጣይቱ ብጡል (፲፰፻፴፪ - ፲፱፻፲) አጭር የሕይወት	ቀኛዝማች ታደሰ ዘወልዴ		

	ታሪክ			
8.	የአግራፊ	ኤሪካሶን	፲፱፻፲፫ ዓ.ም	አስመራ በሚሲኦንግ ስዌዴክ፣ ታተመ
9.	ከወልወል እስከ ማይጨው	ብርሃኑ ድንቄ	መስከረም ፳፮ ቀን ፲፱፻፵፪ ዓ.ም	ትንሣኤ ዘጉባኤ ማተሚያ ቤት
10.	ልጅነት ተመልሶ አይመጣም	እምነት ገብረ አምላክ	፲፱፻፶፱ ዓ.ም	ትንሣኤ ዘጉባኤ ማተሚያ ቤት
11.	ማርክሲዝምና የቋንቋ ችግሮች	ጆሴፍ እስታሊን ትርጉም፣-በደበበ ሰይፉ	፲፱፻፸ ዓ.ም.	ሴንትራል ማተሚያ ቤት
12.	መሬት የማን ነው?	አቤ ጉበኛ	፲፱፻፷፯ ዓ/ም	ትንሣኤ ዘጉባኤ ማተሚያ ቤት
13.	ትቤ አካሉም መኑ አንተ?	አስረስ የኔሰው	በ፲፱፻፶፩ ዓ/ም	ንግድ ማተሚያ ቤት
14.	ጦቢያ	አፈ ወርቅ ገ/አ.የሱስ	1900 ዓ.ም.	ሮማ ከተማ
15.	የአቤቶሁን ያዕቆብ ትውልድና አጭር ታሪክ	ታደሰ ወልዴ	አዲስ አበባ ፲፱፻፵፰ ዓ.ም	ብርሃንና ሰላም የቀዳማዊ ኃይለ ሥላሴ ማተሚያ ቤት
16.	የዓለም መስተዋት	አሐዱ ሳቡሬ	የካቲት ፲፰ ቀን ፲፱፻፵፮ ዓ.ም	
17.	የዓመፅ ኑዛዜ	አቤ ጉበኛ	መስከረም ፱ ቀን ፲፱፻፶፭ ዓ.ም	ብርሃንና ሰላም ማተሚያ ቤት
18.	የማይጨው ቁስለኛ	መኰንን ዘውዴ	ጥቅምት ፳፫ ቀን ፲፱፻፵፰ ዓ.ም	ብርሃንና ሰላም ማተሚያ ቤት

Table 5.1 Books which are written in Amharic language

NO	Title	Author	Publication year	Place of publication
1.	ሮቢንሶን ክሩስ	ዳንየል ደፎ ትርጉም፡ በሙሳ አሮን	መስከረም ፲፱፻፶ ዓ.ም	አስመራ

Table 5.2: Book which are written in Tigrigna language

NO	Title	Author	Publication year	Place of publication
1.	ምዕራፍ	ቅዱስ ያሬድ		
2.	መፀሀፈ ቅዳሴ			

3.	ጸመ ድን	ቅዱስ ያሬድ	ከ፳፻፵ እስከ ፳፻፰	ትንሣኤ ዘጉባኤ ማተሚያ ቤት
4.	የቅዱስ ያሬድ ታሪክና የዜማው ምልክቶች	ሊቀ ጠበብት አክሊለ ብርሃን ወልደ ቂርቆስ እና ሌሎችም	ግንቦት ፲፭ ቀን ፲፱፻፶፱ ዓ.ም.	ትንሣኤ ዘጉባኤ ማተሚያ ቤት
5.	ዝማሬ ወመዋሥዕት	እደ መዝገቡ		
6.	መጽሐፈ ዚቅ ወመዝሙር	ቀለመ ወርቅ ለውጤ ዘብሔረ ጎንደር ወስመ ደብሩ አሸማ ቂርቆስ		ትንሣኤ ዘጉባኤ ማተሚያ ቤት
7.	ጥንታዊ ሆሄ ዘ ልሳነ ግዕዝ	አባ ተ/ማርያም	አመ ፲፮ ለመስከረም ፲፱፻፹፮ ዓ.ም.	ኢየሩሳሌም
8.	መጽሐፍ ሰዓታት	ሊቀ ሊቃውንት አባ ጥዑመ ልሳን ኪ/ማርያም		

Table 5.3: Book which are written in Geez language

Other books of Geez which are used in this thesis are Wisdom of Solomon, Testament of Adam, Kebre Negest and etc.

NO	Title	Author	Publication year	Place of publication
1.	ሺንጋ ጨነዊም።	ዳንየል ደፎ	መስከረም ፲፱፻፶፯ ዓ.ም	አስመራ
2.	ተነ ቅማርም ቅራጭም ትኻርም			
3.	የጫሙት ሸካ			

Table 5.4: Book which are written in Guragigna language

5.2.4 News

We use different news web sites like EBC, FBC, and ESAT etc. The data from news web sites covers domains such as health, agriculture, sport, business, social, politics, and sport. Hence the corpus spans several domains. We found for Tigrigna and Amharic per language the size of collected text corpus are 193,925 words for Tigrigna and 59,219 words for Amharic languages.

5.2.5 Data Cleaning

For training corpora: - Data cleaning for training involves removing of numbers, special characters and mathematical symbols. These cleanings are done automatically which is developed in Java program in preprocessing step. Table 5.5 shows the corpus sizes of the languages after data cleaning.

Languages	Corpus Size per words
Amharic	918,792
Tigrigna	754,957
Geez	540,847
Guragigna	21,489

Table 5.5: total Corpus size after data cleaning

The final document collection used in this thesis consists of about 2,236,085 words across 3 main sources, totaling over 27 Mb of data in 4 languages are used.

5.3 Cross-Validation

Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is k-fold cross-validation. Other forms of cross-validation are special cases of k-fold cross-validation or involve repeated rounds of k-fold cross-validation.

In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed, such that each iterations in a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning.

In machine learning 10-fold cross-validation ($k = 10$) is the most common. Cross-validation is used to evaluate or compare learning algorithms as follows: In each iteration, one or more learning algorithms use $k-1$ folds of data to learn one or more models, and subsequently the learned models are asked to make predictions about the data in the validation fold. The performance of each learning algorithm on each fold can be tracked using some predetermined performance metric like recall, Precision etc. Upon completion, k samples of the performance metric are available for each algorithm. Different methodologies such as averaging can be used to obtain an aggregate measure from those samples, or these samples can be used in a statistical hypothesis test to show that one algorithm is superior to another.

Now the issue is how to select an appropriate value for k . A large k is seemingly desirable, since with a larger k (i) there are more performance estimates, and (ii) the training set size is closer to the full data size, thus increasing the possibility that any conclusion made about the learning algorithm(s) under test will generalize to the case where all the data is used to train the learning model. As the value of k increases, however, the overlap between training sets also increases. For example, with 5-fold cross-validation, each training sets share only $3/4$ of its instances with each of the other four training sets whereas with 10-fold cross validation, each training sets share $8/9$ of its instances with each of the other nine training sets. Furthermore, increasing k shrinks the size of the test set, leading to less precise, less fine-grained measurements of the performance metric. For example, with a test set size of 10 instances, one can only measure accuracy to the nearest 10%, whereas with 20 instances the accuracy can be measured to the nearest 5%. These competing factors have all been considered and the general consensus in the machine learning community seems to be that $k = 10$ is a good compromise. This value of k is particularly attractive because it makes predictions using 90% of the data, making it more likely to be generalizable to the full data. In this case our thesis stick on this method.

After data cleaning 10 percent of the corpus used to evaluate the models for each of the languages. The remaining 90 percent held for training the models. Table 5.6 and 5.7 shows the average size of training and testing corpus respectively. These sizes of language corpus used to train and test the proposed general purpose language identifier by extracting sequence of character ngrams.

Languages	Corpus Size per words
Amharic	826,836.2
Geez	486,761.7
Guragigna	19,285.2
Tigrigna	679,440.60

Table 5.6: Corpus size for training the models (90%)

Languages	Corpus Size per words
Amharic	91,955.8
Geez	54,085.3
Guragigna	2,203.8
Tigrigna	75,516.4

Table 5.7: Corpus size for testing the models (10%)

For word based Ngram approach we built word list-based models from a training set obtained from 9/10 of the training set of each language. Thus, for each language, we built a word based Ngram model that consisted of the vocabulary derived from the unique set of words in the available data set (9/10 of the dataset) which is done automatically. The other 1/10 of the data set was reserved for testing. The words were organized into a sorted set such that each word featured once in the model. The words were further indexed by word length to improve searching time of the system. Models were labelled by the name of the language from which the text to build the model was derived. After developing the training words the size of corpus used in word based approach is show in table 5.8.

Languages	Corpus Size per words
Amharic	155,847
Tigrigna	98,417
Geez	77,092
Guragigna	8,617

Table 5.8: Corpus size for word based approach

Hence, in this investigation ten iterations were used to conduct the experiments and for each iteration we isolated one part of the dataset for testing while retaining the remaining nine parts as the training set. Then we obtained the accuracy for this first iteration and we repeated the steps for the 2nd to the 10th iterations resulting in accuracy.

Tests	Amharic # word	Geez # word	Guragigna # word	Tigrigna # word
Test 1	91,655	55,070	2,134	78,726
Test 2	92,870	54,244	2,171	77,366
Test 3	91,443	54,421	2,221	78,205
Test 4	91,023	52,524	2,219	75,226
Test 5	92,786	54,332	2,182	75,233
Test 6	91,884	57,063	2,202	77,129
Test 7	91,074	51,556	2,052	75,430
Test 8	91,470	52,632	2,207	72,412
Test 9	92,794	53,300	2,258	72,404
Test 10	92,559	55,711	2,392	73,033
Average	91,955.8	54,085.3	2,203.8	75,516.4

Table 5.9: Statistics of test data corpus

5.4 Implementation

The development tool selected for the proposed approach was java programming language, which is an object oriented programming. Subsequently, among different benefits of object oriented in comparison with other programming paradigms is its simplicity 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 instance experiment 4 was conducted by the procedure of incorporating location feature set along with experiment 3 features and components. Hence, the procedure used in all experiment of this investigation is almost the same; the only difference is the class they access.

5.5 Evaluation

5.5.1 Evaluation Metrics

Evaluation of the proposed language identification is done with the evaluation metrics that compares the number of words which are labelled the language category correctly and incorrectly. In order to achieve this, the language labelling for each test document words are done manually, since this manual language labelling helps for checking the final result of our general purpose language identifier.

Among the different methods that are used to evaluate the performance of a text language identification system, in this investigation in order to measure the effectiveness, quality and performance of proposed approach we adopt the Precision (P), Recall (R) and F-measure (F-m) evaluation parameters.

As explained in chapter 2, equations that were discussed in section of 2.4 were used to compute these evaluation parameters (i.e. Precision, Recall and F-measure).

5.5.2 Test Result

In this work, six experimentation techniques were proposed to observe the strength of our proposed language identifier from deferent angle and perspective. Therefore, the procedure to undertake the whole experiment is the same, i.e. the experiment was undertaken by including or not to including different approaches according to the given experimentation techniques. Through all experiments of the proposed general purpose language identifier, the word ngram as well as the rule based approach are incorporated.

Beside this, in order to show the performance comparison between six experimentation techniques of this investigation we used the monolingual setting document level as test input. However, finally the best performed experimentation techniques are examined in both monolingual as well as multilingual setting documents at four different test levels: word, phrase, sentence and document. In general, in this investigation the following six experimentation techniques are undertaken.

Experiment 1:- Investigating the performance of fixed length character Ngram in LangID

The objective of this experiment is to observe the performance of fixed length character ngram approach along with word ngram and rule based approach. In this experiment the character ngram size range from 2 to 5 are conducted. As shown in below Table 5.10, which depicts the average experimental result of this experiment, in order to make suitable our table based experimental illustration we present the average value of each evaluation metrics (Recall, Precision and F-measure).

Ngram size	Amharic			Geez			Guragigna			Tigrigna		
	Avg. R	Avg. P	Avg. F-m	Avg. R	Avg. P	Avg. F-m	Avg. R	Avg. P	Avg. F-m	Avg. R	Avg. P	Avg. F-m
2ngram	79.59	57.12	66.25	80.68	79.50	79.30	59.20	83.52	62.93	76.71	83.74	76.04
3ngram	77.08	57.57	65.59	82.08	78.79	79.72	58.01	82.23	67.49	76.81	75.24	75.62
4ngram	77.53	60.60	67.74	83.1	79.44	80.59	59.24	82.05	68.33	70.86	75.94	76.82
5ngram	77.87	62.28	68.99	78.46	79.85	78.24	62.31	82.03	70.27	78.05	76.78	77.11

Table 5.10: Experimental Results for experiment 1

Experiment 2:- Investigating the effect of location feature set in fixed length character Ngram

The major purpose of this experiment is to measure the performance of language identification when a location feature set is incorporated in fixed length character ngram. Beside this, the experiment also aims to observe the capability of location feature set incorporated along with fixed length character ngram.

Hence, this experiment is conducted to see the impact of location feature set on the performance of the previous experimentation technique. As a result the proposed language identification with this feature set achieves the experimental results as depicts in below Table 5.11.

Like experiment 1, in order to make easy and suitable our table based illustration of this experiment , we only present the average result of experimental evaluation measurement results (Precision , Recall and F-measure) as shown in below Table 5.11.

Ngram size	Amharic			Geez			Guragigna			Tigrigna		
	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	
	R	P	F-m	R	P	F-m	R	P	F-m	R	P	F-m
2ngram	76.57	59.86	67.37	78.80	75.88	76.87	67.80	75.14	69.49	70.49	81.44	75.16
3ngram	76.21	62.61	68.20	75.93	75.25	74.91	69.96	73.77	70.31	72.41	82.55	76.92
4ngram	76.85	64.32	70.39	70.1	76.8	76.95	69.22	70.22	73.62	74.51	81.60	77.57
5ngram	77.10	65.99	70.38	80.52	78.56	79.15	69.81	79.26	73.69	67.8	83.11	78.98

Table 5.11: Experimental Results for experiment 2

Experiment 3:- Investigating the performance of infinite character Ngram in LangID

This experiment aims to show the performance of the language identification using infinity ngram, which is a combination of different character ngram sizes instead of a fixed one. Moreover, this experiment is conducted in order to observe the effect of infinity ngram over the previous fixed character ngram based language identification. Hence, in this experiment like pervious experiments the infinity ngram is used in combination of word ngram as well as rule based approach of the language identification.

The main intention of this experiment is in order to observe the capability of infinity ngram in our proposed general purpose language identification approach. As well, the experimental result of this experiment is shown in below Table 5.12.

Tests	Amharic			Geez			Guragigna			Tigrigna		
	R	P	F-m	R	P	F-m	R	P	F-m	R	P	F-m
Average	82.67	83.80	83.07	80.4	86.17	82.95	93.70	81	86.67	85.06	86.24	85.53

Table 5.12: Experimental Results of infinite length ngram

Experiment 4:- Investigating the effect of location feature set in infinite character Ngram

The objective of this experiment is to illustration the performance of infinity ngram with combination of location feature set, a location in which a character ngram is occurred in a word.

On the other hand, in order to observe the capability of location feature set in our infinity ngram based language identification. The experimental result of this experimental technique is illustrated in below Table 5.13.

Tests	Amharic			Geez			Guragigna			Tigrigna		
	R	P	F-m	R	P	F-m	R	P	F-m	R	P	F-m
Average	85.54	86.50	85.96	84.73	87.99	86.26	91.83	85.21	88.25	88.49	89.19	88.75

Table 5.13: Experimental Results of infinite length ngram with location feature

Experiment 5:- Investigating the effect of sentence level reformation in Experiment 4

The purpose of this experiment is to observe the performance of language identification used for experiment 4 with inclusion of sentence level reformation module. On the other hand, in order to observe the capability of sentence level reformation module in the proposed general purpose language identification used in experiment 4. Ultimately, the experimental result of this experimental technique is illustrated in below Table 5.14.

When a language dominance of a particular language attains the experimental defined threshold value, then the previous initial language category result is re-improved with dominant language category as category of all words with in the sentence. Through experiment for all supported language of this investigation a benchmark value with 80% language dominance with in a sentence achieves a better result ash shown in above 5.14.

In other words, in order to perform sentence level language re-improvement at least 80% of the words in a sentence must be confirmed or valid , unless the initial language category result is taken as a final language category of all words of a sentence.

Tests	Amharic			Geez			Guragigna			Tigrigna		
	R	P	F-m	R	P	F-m	R	P	F-m	R	P	F-m
Average	100	99.7	99.85	99.62	99.85	99.74	100	100	100	99.87	100	99.93

Table 5.14: Experimental Results of sentence reformation feature

Experiment 6:- Investigating the effect of document level reformation in Experiment 5

The major intention of this experiment is to measure the performance of our general purpose language identification system with inclusion of document level reformation along with the experiment 5 components. In other words, during this experiment the location feature set is combined with sentence level as well as document level reformation in order to observe the effect of this combination.

Unlike the pervious experimental techniques, this experiment is conducted a both monolingual as well as multilingual setting in four different levels: word, phrase, sentence and document level in order to observe the strength and effectiveness of the language identification. In order to make the experimental result of this experiment we illustrate the average value of evaluation parameters (i.e. Recall, Precision and F-measure) in both monolingual as well as multilingual setting along four previous levels.

A. Experiment Result at Monolingual Setting

Table 5.15 depicts the experimental result for language identification with combination of location based infinity ngram with sentence level as well as document level reformation at a monolingual setting.

Test level	Amharic			Geez			Guragigna			Tigrigna		
	Avg. R	Avg. P	Avg. F-m	Avg. R	Avg. P	Avg. F-m	Avg. R	Avg. P	Avg. F-m	Avg. R	Avg. P	Avg. F-m
Phrase	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Sentence	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Document	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table 5.15: Experimental Results of monolingual texts

The document level dominance threshold value adjustment with 95% value is best performed language identification for all supported Ethiopian Semitic languages. In other words , this module adjust incorrect labelling of monolingual documents in to more languages through adjusting the sentence level reformed language category result into a single dominant language category when a particular language satisfied the document dominance threshold value.

B. Experiment Result at Multilingual Setting

Since our proposed language identifier aims to labeling the language category at a word level, hence it is out performed for language categorization of textual documents written in more than one language. Table 5.16 illustrates the experimental result of such features of our general purpose language identifier, a combination of infinity ngram with sentence level and document level reformation proses at a multilingual setting test documents.

Test level	Amharic			Geez			Guragigna			Tigrigna		
	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.	Avg.
	R	P	F-m	R	P	F-m	R	P	F-m	R	P	F-m
Phrase	84.40	81.29	82.64	81.55	92.57	86.38	95.71	80.86	87.19	84.44	90.19	86.81
Sentence	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Document	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table 5.16: Experimental Results of multilingual texts

5.6 Discussion

As shown in Table 5.10, the experimental result in average character ngram with size 5 is the best performing setting for language identification task of supported Ethiopian Semitic language. For character ngram size 5 setting we reach an average F-measure of 68.99 %, 78.24%, 70.27% and 77.11 % for Amharic, Geeze, Guragigna and Tigrigna respectively. However, particularly for Geeze language we reach a better F-measure value (83.1%) for 4 character ngram size setting. As shown from this experiment, in average the language identification for small character ngram size is lower than the result reported for higher character ngram size. This indicate that , for documents written with Ethiopian Semitic language with lower character ngram size , the character occurrence pattern is more similar and is not performed well character ngram size setting to discriminate the language of given input text document.

On the other hand, as shown in Table 5.11 we observe that incorporating the location of characters ngram in a word feature set achieves a better of an overall experimental result. Like, experiment 1 the average character ngram with size 5 is the best performing setting for language identification task of this experimental technique. As shown from experimental result in above Table 5.11, the performance of language identifier with character ngram size of 5 setting, it increases with 1.39%, 0.91%, 3.42% and 1.87 for Amharic, Geeze, Guragigna and Tigrigna respectively. This is because of that the Ethiopian sematic languages used in this investigation has differ to each other based on the occurrence character ngram location in a word, in other words a particular characters and sequence of tendencies are occurred at the particular location

pattern frequently. However, the experimental result for experiment 1 as well as experiment 2 shows lower performance of the language identification and this is due to extraction of small number of features in order to perform decision of language labeling. Since, the language labelling is done a word level and in order to decide the language category the features for a fixed size of character ngram is used and this is not much enough in order to perform a decision of a language category of a word.

From the result of Experiment 3 , as shown in Table 5.12 Amharic, Geeze, Guragigna and Tigrigna achieves an average F-measure of 83.07%, 82.95%, 86.67%, and 85.53% respectively. when we compare the performance of this experiment 3 with previous best performed experiment 2 , we observe that Amharic , Geeze , Guragigna and Tigrigna increases its language identification performance with average F-measure of 12.69% , 4.39% , 7.41% and 6.55% respectively. The performance enhancement implies that, the infinity ngram approach is a strength and efficient language identification technique than that of the fixed character ngram approach. This is because of that, during infinity ngram the features extracted to represent the individual words is larger and larger than the fixed character ngram, when reach features are given to a language identifier to decide the language category of a word a better accuracy is found.

Beside this, as shown in Table 5.13, the experimental result of experiment 4 scores 85.96%, 86.26%, 88.25% and 88.75% for Amharic, Geez, Guragigna and Tigrigna respectively when a location feature set is incorporated along with infinity ngram. From the result of this experiment, the performance of the language identification when compared with previous experiment 3 result, it increases with 2.89%, 3.31% , 1.58% and 3.22% for Amharic , Geeze , Guragigna and Tigrigna respectively. This performance enhancement implies that, the incorporation of location feature set along with infinity ngram based language identification is a best combination setting for language identification of Ethiopian Semitic languages. Similarly as explained earlier, this is due to the factor in which the characters occurrence pattern in a word is particularly unique from one to another language of Ethiopian Semitic languages. In other words , location feature set or position occurrence of character ngram within a word is a best feature set in order to discriminate a these languages, particularly when incorporated with infinity ngram based approach.

As shown Table 5.14, the more performance of language identification is recorded when compared with all previously done experimental techniques of this investigation. In another words, this best language identification performance indicates the capability of sentence level reformation in our general purpose language identification task. Hence, as shown in above Table 5.14, the experimental result achieves 99.85%, 99.74%, 100% and 99.93% for Amharic, Geeze, Guragigna and Tigrigna respectively. When compare this experimental result with experiment 4, which means before the sentence level reformation is incorporated, the performance of the language identification is increase with 13.89%, 13.48%, 11.75% and 11.18% for Amharic, Geeze, Guragigna and Tigrigna respectively. This extreme performance enhancement implies that inclusion of sentence level reformation along with location feature set based infinity ngram is a best combination setting of the proposed language identification. The reason behind the extreme performance enhancement of language identification during the inclusion of sentence level reformation module with respect to all the languages which are used for experiment is due to the capability of this module , which is looking an scanning the sentence level language dominance.

From the experimental result as shown in Table 5.15, we observed that the sentence level language identification is achieve an average of 100% F-measure for all supported Ethiopian Semitic languages and this is due to the inclusion of sentence level reformation, since it enhances the initial language labelling through re-adjusting when the language dominance threshold value is confirmed (as explained in a previous experiment 5).On the other hand, the experimental result of this experiment at a document level of monolingual setting also achieves an average of F-measure 100% for all supported Ethiopian Semitic language and this is due to the incorporation of document level reformation module. Since, this module adds an improvement on a language category of a sentence level reformation at a document level as a whole, when the given test document confirm at least the experimental threshold value called document level dominance threshold value.

As well, as shown the experimental result of this experiment at monolingual setting for phrase level test document, also achieves an average F-measure of 100% for all supported languages. The reason behind this experimental result is that, in our proposed general purpose language identification the test levels with are smaller in size than the sentence level are considered as a

single sentence level test document. Hence, like that of sentence level test document the re-adjustment for such inputs (i.e. phrase level) is done, when the sentence level language dominance threshold value is confirmed. As we have seen in a previous experiment 4 the combination of location feature set based infinity ngram with word ngram as well rule based approach performs well and the performance always achieves the sentence level dominance threshold value. Hence, the sentence level re-improvement is always done for the phrase level initial language category result.

Similarly like experimental result of experiment 4 at monolingual setting documents, at sentence level as well as document level test at multilingual level scores an average F-measure of 100% for all supported Ethiopian Semitic languages as shown in Table 5.16. However, the document level reformation module has not any factory during this multilingual document setting, since the test document is multilingual not reach the document level dominance value (95%). Hence, during the language identification of multilingual document setting, the sentence based re-adjustment plays a great role in order to identify the language category at sentence as well as document level multilingual textual test documents. On the other hand, as we have seen from the above experimental result Table 5.16 of multilingual document setting, the performance of the language identification achieves an average F-measure of 82.64%, 86.38%, 87.19% and 86.81% For Amharic, Geeze, Guragigna and Tigrigna respectively. As the result of this experiment indicates the performance of language identification decreased with 17.36%, 13.62%, 12.81% and 13.19% for Amharic, Geeze, Guragigna and Tigrigna respectively from sentence and document test level.

This is because of that, at phrase level there is no re-adjustment of the language category result, since the threshold value is not fulfil the sentence level dominance threshold value. Hence, the initial language category of words in a phrase is taken as a final re-adjusted language category. As we explained earlier, without sentence level adjustment the performance of the language identification is not 100% accurate , since the languages are very closely related due to that the language identification is challenged to discriminate words language category exactly like that of done for sentence level and document level.

Chapter Six

Conclusion and Recommendation

6.1 Conclusion

The textual documents written in different language getting more and more available on the global network and in order to use this content of textual document for further processing, one should know the language in which it is written. Therefore, an automatic language identification mechanism is required in order to identify the language category in which the textual document content is written.

In order to solve this research problem, in past decades a number of research works have been conducted in the area of language identification. However, there are three main issues which makes the language identification is still hot research area or unsolved problem: identifying language for multilingual textual documents, identifying language for very closely related languages and also identifying language category for very short texts like words or phrases.

Hence, in order to solve such language identification difficulty this investigation presented a general purpose language identification approach. This proposed approach able to identify the language of textual document in both monolingual as well as multilingual setting. As we had seen from the experiment of this investigation, this proposed approach also able to classify the textual documents language category in four different levels of test document: word, phrase, and sentence and document level. The reason behind this capability of the proposed approach is the language identification is done on an individual word basis or word level.

Moreover , even if our language identifier is able to identify the language of a document in any document setting (i.e. monolingual as well as multilingual) , but in order to train the language identifier no need of multilingual dataset rather it needs only any monolingual row text of all supported language. Hence, in order to extend the proposed language identifier to other languages needs only the row text of a target language.

In this investigation, the corpus of each language is divided into two (training set and testing set) for testing and training purpose. We have two training corpus for character and word ngram. The training set for character ngram consists 90% of the corpus and the testing set consists 10% of the corpus. For word ngram we use 155847, 98417, 77092 and 8617 of unique words for Amharic, Tigrigna, Geez and Guragigna respectively.

As explained before, in this investigation we conduct six experimental techniques through combining the different approaches of the proposed general purpose language identifier. In all experimental techniques the word ngram and rule based approach used as a default component of the language identification. The experimental result is evaluated based on basic evaluation metrics: precision, recall and F-measure. Ultimately, based on the experimental result the combination of location feature set in infinity ngram with sentence level reformation and document level reformation performed better and attains an average F-measure of 100% for word, phrase, and sentence and documents level in a monolingual setting. As well, for multilingual setting also attains an average F-measure of 100% for both sentence level and document level test, but for phrase level achieves an average F-measure of 82.64%, 86.38%, 87.19% and 86.81% For Amharic, Geeze, Guragigna and Tigrigna respectively.

6.2 Contribution

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

- A generic model is proposed for Language identification which can identify any level of text (i.e. Word, phrase, sentence and document level)
- Proposed a word level language identifier which performs better in both monolingual as well as multilingual document settings. For multilingual language identification the switching point of a language category is detected accurately without confusion which is not considered in a previous works.
- Proposed an approach which can identify very closely related languages (i.e. Amharic, Geeze, Guragigna and Tigrigna) with better performance.
- Proposed an approach which uses contextual information of a word in order to disambiguate an ambiguous word of a language. Since, in Ethiopian Semitic language there are words which are common in more than one language and can be labeled to one of a language category based on its contextual information.
- A location feature set is used in combination with character ngram probability in order to enhance the performance of the proposed word level language identifier.
- Used a new approach called infinity ngram, which uses a combination of all character ngrams to enhance the number of features extracted per word. An infinity ngram approach also used in combination with location feature set to observe the effect on a proposed approach.
- The study also contributes on the improvement of language identification by proposed a general purpose language identifier that operates at word level.
- In addition, the study contributes to the growth of Ethiopian Semitic languages accessibility on different NLP application. Since, identify a language is a pre-processing task in order to perform other higher NLP related tasks.

6.3 Recommendation

In this investigation we described a general purpose language identifier, which able to identify the language category of a text at any level (i.e. word, phrase, sentence and document) in both monolingual and multilingual setting. The result found from this research showed that, the proposed approach very well in identifying the language category of any level of a text and in any language setting. Finally, the researcher recommends the following points as further research directions.

- During this investigation we used a character ngram frequency and location feature set in order to represent the features of both training and testing documents. However, it would be better to use more sophisticated features technique, which able to select features that are more likely to discriminate between very closely related languages.
- In this investigation we adopt Laplace smoothing, which is simply adding one count across a data set. However, there are different better performed smoothing techniques that can be applied to the language identification tasks such as Katz smoothing, Witten-Bell smoothing , absolute discounting, Kneser-Ney discounting and Jelinek-Mercer in order to enhance the performance of the proposed general purpose language identification.
- In this investigation, we adopt a bayesian classifier for the final classification module of the proposed general purpose language identification. However, in order to observe the best machine learning classifier along our proposed language identifier it is recommended that to perform a comparative study with other different machine learning classifier.
- From the experiment that has been conducted, the infinity ngram based approach is outperforming than fixed size character ngram approach for language identification task at both monolingual as well as multilingual document setting. Having this in mind, the researcher recommends adopting this new approach for other classification tasks.

References

- [1] Lewis, M. Paul, Gary F. Simons, and Charles D. Fennig (eds.). (2014). *Ethnologue: Languages of the World*, Seventeenth edition. Dallas, USA: SIL International. Online version: <http://www.ethnologue.com>. [Accessed: 19- Dec- 2016].
- [2] Hughes Baden, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew MacKinlay, (2006). “Reconsidering language identification for written language resources“, *In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Italy, Genoa, 485–488, pp.
- [3] Marco Lui, Jey Han Lau and Timothy Baldwin. (2014). “Automatic Detection and Language Identification of Multilingual Documents”, *Transactions of the Association for Computational Linguistics*, pp. 27–40.
- [4] Teklehaimanot, T. (2002) *Ethiopian languages - Semitic, Cushitic, Omotic and Nilo-Saharan*. Available at: <http://www.ethiopiantreasures.co.uk/pages/language.htm> (Accessed: 19 December 2016).
- [5] Ronny Meyer. (2016) “Amharic as lingua franca in Ethiopia”, Lissan: *Journal of African Languages and Linguistics*. Available at: [http://www.academia.edu/5514187/Amharic as lingua franca in Ethiopia](http://www.academia.edu/5514187/Amharic_as_lingua_franca_in_Ethiopia). [Accessed 3 December 2016].
- [6] Judith Rosenhouse. (2013). *Globally Speaking: Motives for Adopting English Vocabulary in Other Languages (Multilingual Matters)*.Pg.165. 1 Edition. Multilingual Matters.
- [7] [Online]. Available: <http://ohr.dc.gov/sites/default/files/dc/sites/ohr/publication/attachments/LAAFactSheet-English.pdf>. [Accessed: 20- Dec- 2016].
- [8] Simons, Gary F. and Charles D. Fenning (eds.). (2017). *Ethnologue: Language of the world*, 20th edition. Dallas, Texas: SIL International. online version: <https://www.ethnologue.com> [Accessed: 20- Jan- 2017].
- [9] Legesse Wedajo Desta. (July 2014). *Modeling Text Language Identification for Ethiopian Cushitic Languages*.
- [10] Sreejith C, Indu M, Dr. Reghu Raj P C. (2013) “N-gram based Algorithm for distinguishing between Hindi and Sanskrit texts”, *Proceedings of the Fourth IEEE International Conference on Computing, Communication and Networking Technologies*, July 4 - 6, 2013.

- [11] Kheireddine Abainia, Siham Ouamour and Halim Sayoud, (2014) "Robust Language Identification of Noisy Texts".
- [12] Yugesh. Sharma, (October 2015). Multilingual Detection System in Indian Languages.
- [13] Fela Winkelmoen, (2010). Statistical Language Identification of Short Texts.
- [14] "Ge'ez language," in Wikipedia, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Ge%27ez_language. Accessed: Dec. 20, 2016.
- [15] En.wikipedia.org. (2017). *Gurage languages*. [Online] Available at: https://en.wikipedia.org/wiki/Gurage_languages [Accessed 28 Jan. 2017].
- [16] Lui and Paul Cook. (2013). "Classifying english documents by national dialect." *In Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013)*, 5–15, Brisbane, Australia.
- [17] Řehůrek R, Kolkus M. (2009) "Language Identification on the Web: Extending the Dictionary Method." *In Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing 2009, Proceedings*. Vyd. první. Mexico City, Mexico: Springer-Verlag, 2009. ISBN 978-3-642-00381-3, pp. 357-368.
- [18] Visa, A.: Technology of Text Mining. In: Perner, P. (ed.) *MLDM 2001. LNCS (LNAI)*, vol. 2123, pp. 1–11. Springer, Heidelberg (2001)
- [19] Beesley, Kenneth R. 1988. "Language Identifier: A computer program for automatic natural-language identification of on-line text." In *Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*, 47–54, Seattle, USA.
- [20] Ronen Feldman and Ido Dagan. (1995) *KDT - Knowledge Discovery in Texts Proceedings of the First International Conference on Knowledge Discovery (KDD)*, 112–117
- [21] Dunning, Ted. 1994. Statistical identification of language. Technical Report MCCS 940-273, Computing Research Laboratory, New Mexico State University.
- [22] McNamee (2005). McNamee, Paul. 2005. "Language identification: a solved problem suitable for undergraduate instruction". *Journal of Computing Sciences in Colleges* 20.94–101.
- [23] Quasthoff, Uwe ; Richter, Matthias ; Biemann, Christian (2006) "Corpus portal for search in monolingual corpora". In: *Proceedings of the fifth international conference on language resources and evaluation* Bd. 17991802
- [24] Cavnar, William B., and John M. Trenkle. (1994). "N-gram based text categorization". *In*

Proceedings of the Third Symposium on Document Analysis and Information Retrieval, 161–175, Las Vegas, USA.

- [25] Muntsa Padro, Lluís PadróCirera, (2004) Comparing methods for language identification, *In Procesamiento del lenguaje natural, Barcelona: Sociedad Española para el Procesamiento del Lenguaje Natural*, pp. 155-161.
- [26] Vojtek, Peter, and Mária Bieliková. (2007). Comparing natural language identification methods based on Markov processes. In *Proceedings of the Fourth International Seminar on Computer Treatment of Slavic and East European Languages (SLOVKO 2007)*, 126–138, Bratislava, Slovakia.
- [27] Grefenstette, Gregory. (1995). Comparing two language identification schemes. *In Proceedings of Analisi Statistica dei Dati Testuali (JADT)*, 263–268, Rome, Italy.
- [28] Elworthy, David. (1998). Language identification with confidence limits. *In Proceedings of the 6th Annual Workshop on Very Large Corpora*, 94–101, Montreal, Canada.
- [29] Hakkinen, Juha, and Jilel Tian. (2001). n-gram and decision tree based language identification for written words. *In Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001. ASRU '01., 335–338, Madonna di Campiglio, Italy.
- [30] Lui, Marco, and Timothy Baldwin. (2011). Cross-domain feature selection for language identification. *In Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, 553–561, Chiang Mai, Thailand.
- [31] Winkelmolen, Fela, and Viviana Mascardi. (2011). Statistical language identification of short texts. *In Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*, 498–503, Rome, Italy.
- [32] Teahan, W. J. (2000). Text classification and segmentation using minimum crossentropy. *In Proceedings of the 6th International Conference Recherche d'Information Assistee par Ordinateur (RIAO'00)*, 943–961, College de France, France.
- [33] Nakatani, Shuyo, (2010). Language detection library. SLangID es. <http://www.slideshare.net/shuyo/language-detection-library-for-java>. Retrieved on 21/03/2017.
- [34] Tian, Jilei, and Janne Suontausta. (2003). Scalable neural network based language identification from written text. *In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, 48–51, Hong Kong.

- [35] Kikui, Genitiro. (1996).” Identifying the coding system and language of on-line documents on the internet.” *In Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, 652–657, Kyoto, Japan.
- [36] TROMP, Erik ; Pechenizkiy, Mykola (2011) Graph-based n-gram language identification on short texts. *In: Proc. 20th Machine Learning conference of Belgium and The Netherlands*, S. 27–34
- [37] Dat Tran and Dharmendra Sharma.” Markov Models for Written Language Identification,” University of Canberra, Australia
- [38] William M. Campbell, Joseph P. Campbell, Douglas A. Reynolds, Elliot Singer and Pedro A. Torres-Carrasquillo. (2006).” Support Vector Machines for Speaker and Language Recognition,” MIT Lincoln Laboratory, Lexington
- [39] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze (2009),” An Introduction to Information, Retrieval,” Cambridge University Press, Cambridge, England
- [40] Carol Peters, Martin. Braschler, Paul Clough. (2012) Multilingual Information Retrieval: From Research to Practice. Springer Science & Business Media
- [41] Hidayet. Takçi, Ekin. Ekin, (2012) Minimal Feature set in language identification and finding suitable classification method with it, in *Procedia Technology* 1, pp. 444-448.
- [42] Gerard Salton, Michael McGill, (1983) *Introduction to Modern Information Retrieval*, McGraw Hill.
- [43] Thorsten Joachims, (2002). *Learning to Classify Text using Support Andctor Machines*, Kluwer, Boston.
- [44] Haykin, Simon, (1998). *Neural Networks: A Comprehensiand Foundation* (2 ed.), Prentice Hall, ISBN 0132733501.
- [45] R. O. Dua, P. E. Hart, D. H. Stork, (2000). *Pattern Classification* (2nd ed.), Wiley Interscience, ISBN 0-471-05669-3.
- [46] van Rijsbergen, C. J. 1979. *Information Retrieval*. London, UK: Butterworths.
- [47] Murthy, Kavi Narayana, and G. Bharadwaja Kumar. 2006. Language identification from small text samples. *Journal of Quantitative Linguistics* 13.57–80.
- [48] Baldwin and Marco Lui. Language identification: The long and the short of the matter. *In Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*,

229–237, Los Angeles, USA.

- [49] Takçı, Hidayet, and Tunga Güngör. 2012. A high performance centroid-based classification approach for language identification. *Pattern Recognition Letters* 33:2077–2084.
- [50] Vogel, John, and David Tresner-Kirsch. 2012. Robust language identification in short, noisy texts: Improvements to LIGA. In *Proceedings of the 3rd International Workshop on Mining Ubiquitous and Social Environments (MUSE)*, 1–9, Bristol, UK.
- [51] Sibun, Penelope, and Jeffrey C. Reynar. 1996. “Language determination: Examining the issues.” In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*, 125–135, Las Vegas, USA.
- [52] Ljubešić, Nikola, Nives Mikelić, and Damir Boras. 2007. “Language identification: how to distinguish similar languages?” In *Proceedings of the 29th International Conference on Information Technology Interfaces*, 541–546, Cavtat, Croatia.
- [53] Tiedemann, Jörg, and Nikola Ljubešić. 2012. “Efficient discrimination between closely related languages.” In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, 2619–2634, Mumbai, India.
- [54] King, Ben, and Steven Abney. 2013. “Labeling the languages of words in mixed language documents using weakly supervised methods.” In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1110–1119, Atlanta, Georgia.
- [55] Nguyen, Dong, and A. Seza Dogruoz. 2013. “Word level language identification in online multilingual communication.” In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, 857–862, Seattle, USA.
- [56] Ling, Wang, Chris Dyer, Alan W Black, and Isabel Trancoso. 2013. “Paraphrasing 4 microblog normalization.” In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 73–84, Seattle, USA.
- [57] Tromp, Erik, and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts. In *Proceedings of Benelearn 2011*, 27–35, The Hague, Netherlands.
- [58] Huang, Chu-Ren, and Lung-Hao Lee. 2008. “Contrastive approach towards text source classification based on top-bag-of-word similarity.” In *Proceedings of the 22nd Pacific Asia*

Conference on Language, Information and Computation, 404–410, Cebu City, Philippines.

- [59] Alex, Amit Dubey, and Frank Keller. 2007. “Using foreign inclusion detection to improve parsing performance.” In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning 2007 (EMNLP-CoNLL 2007)*, 151–160, Prague, Czech Republic.
- [60] Cook, Paul, and Marco Lui. 2012. “langid.py for better language modelling.” In *Proceedings of the Australasian Language Technology Association Workshop 2012*, 107–112, Dunedin, New Zealand.
- [61] Scannell, Kevin P. 2007. The Crúbadán Project: Corpus building for under resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, 5–15, Louvain-la-Neuve, Belgium.
- [62] Abney, Steven, and Steven Bird. 2010. The human language project: building a universal corpus of the world’s languages. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, 88–97, Los Angeles, USA.
- [63] Resnik, Philip. 1999. “Mining the Web for bilingual text.” In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 527–534, College Park, USA.
- [64] Nie Michel Simard, Pierre Isabelle, and Richard Durand. 1999. “Cross language information retrieval based on parallel texts and automatic mining of parallel texts from the web.” In *Proceedings of 22nd International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR’99)*, 74–81, Berkeley, USA.
- [65] Yamaguchi, Hiroshi, and Kumiko Tanaka-Ishii. 2012. “Text segmentation by language using minimum description length.” In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, 969–978, Jeju Island, Korea.
- [66] Brown, Ralf. 2013. “Selecting and weighting n-grams to identify 1100 languages.” In *Proceedings of the 16th International Conference on Text, Speech and Dialogue (TSD 2013)*, 475–483, Plzeň, Czech Republic.
- [67] Peng, Fuchun, Fangfang Feng, and Andrew McCallum. 2004. “Chinese segmentation and new word detection using conditional random fields.” In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, 562–568, Geneva,

Switzerland.

- [68] Anil Kumar Singh and Jagadeesh Gorla. Identification of Languages and Encodings in a Multilingual Document {anil,jagadeesh}@research.iiit.ac.in Language Technologies Research Centre IIIT, Hyderabad, India
- [69] Ranaivo-Malancon, Bali. 2006. Automatic identification of close languages – case study: Malay and Indonesian. *ECTI Transaction on Computer and Information Technology* 2.126–134.
- [70] Sites, Dick, 2013a. Cld2fullversion. Online manuscript. Available at <http://code.google.com/p/cld2/wiki/CLD2FullVersion>.
- [71] Zampieri, Marcos. 2013. “Using bag-of-words to distinguish similar languages: How efficient are they?” *In Proceedings of the 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, 37–41, Budapest, Hungary.
- [72] Abdelmalek Amine, ZakariaElberrichi, Michel Simonet, (2010), “Automatic Language Identification: An Alternative Unsupervised Approach Using a New Hybrid Algorithm”, *IJCSA* 7(1), 2010, pp. 94-107.
- [73] Gordon, R.G., 2005. *Ethnologue: Languages of the world*. SIL International, Dallas, TX.
- [74] Da-Silva, J.F., Lopes, G.P., 2006. “Identification of document language is not yet a completely solved problem.” *In: Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce, IEEE Computer Society*, p. 212.
- [75] Prager, John M. 1999. “Linguini: language identification for multilingual documents.” *In Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences (HICSS-32)*, Maui, USA.
- [76] Peirsman, Yves, Dirk Geeraerts, and Dirk Speelman. 2010. The automatic identification of lexical variation between language varieties. *Natural Language Engineering* 16.469–491.
- [77] Trieschnigg, Dolf, Djoerd Hiemstra, Mariët Theune, Franciska Jong, and Theo Meder. 2010. “An exploration of language identification techniques for the dutch folktale database.” *In Proceedings of the LREC workshop Adaptation of Language Resources and Tools for Processing Cultural Heritage*, 2 6, Istanbul, Turkey.
- [78] Diwersy, Sascha, Stefan Evert, and Stella Neumann. 2014. “A weakly supervised

multivariate approach to the study of language variation.” In *Aggregating Dialectology, Typology, and Register Analysis. Linguistic Variation in Text and Speech*, ed. by Benedikt Szmrecsanyi and Bernhard Wälchli. Berlin: De Gruyter.

- [79] Elfardy, Heba, and Mona Diab. 2013. “Sentence level dialect identification in Arabic.” *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 456–461, Sofia, Bulgaria.
- [80] Zaidan, Omar F, and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics* 40.171–202.
- [81] Mandl, Thomas, Margaryta Shramko, Olga Tartakovski, and Christa Womser-Hacker. 2006. “Language identification in multi-lingual webdocuments.” *In Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems (NLDB 2006)*, 153–163, Klagenfurt, Austria.
- [82] Zampieri, Binyam Gebrekidan Gebre, and Holland Nijmegen. 2012. “Automatic identification of language varieties: The case of Portuguese.” *In Proceedings of The 11th Conference on Natural Language Processing (KONVENS 2012)*, 233–237, Vienna, Austria.
- [83] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.
- [84] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” in *Proc. ACL*, 1996, pp. 310–318.
- [85] S. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” in *Proc. ICASSP*, 1987, pp. 400–401.
- [86] Mahmudul Hasan, Saria Islam, and Arifur Rahman, “A comparative study of Witten Bell and Kneser-Ney smoothing methods for statistical machine translation,” *Journal of Information Technology*, vol. 1, pp. 1–6, 2012.
- [87] Hermann Ney, Ute Essen, and Reinhard Kneser, “On structuring probabilistic dependences in stochastic language modelling,” *Computer Speech & Language*, vol. 8, no. 1, pp. 1–38, 1994.
- [88] Frederick Jelinek, “Up from trigrams,” in *Proc. EUROSPEECH*, 1991, pp. 1037–1040.
- [89] Disuke O. Jun I.T, 2009. Text Categorization with All Substring Features.
- [90] Meron Sahlemariam , Libsie M. and Yacob D., 2009, "Concept-Based Automatic Amharic

Document Categorization", in Americas Conference on Information Systems (AMCIS).

[91] Tesfaye Tewelde (PhD), (2002). A modern grammar of Tigrigna, Tipografia U. Detti – via G. Savonarola Roma.

[92] Tigrigna Alphabet and pronunciation: <http://www.omniglot.com/writing/Tigrigna.htm>: visited on August 26, 2017.

[93] Tigrigna Grammar, (1996). American Evangelical Mission, First Red Sea Press, Inc., Edition.

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.

Kidst Ergetie Andargie

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



Yaregal Assabie, PhD
Addis Ababa

Jimma, Ethiopia November 2017