Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Machine Learning Approach for Morphological Analysis of Tigrigna Verbs

A Thesis Submitted to Addis Ababa Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

By
Gebrearegay Kalayu Abraha

October, 2018

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Machine Learning Approach for Morphological Analysis of Tigrigna Verbs

By
Gebrearegay Kalayu Abraha

Advisor

Dr. Eng. Getachew Alemu

Declaration

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

Declared by: Gebrearegay Kalayu Abraha

Signature:----------

Date: ----------

Place: Addis Ababa institute of Technology, Addis Ababa University, Addis Ababa

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

Confirmed by:

Advisor's Name: Dr.Eng. Getachew Alemu

Signature:----------

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Machine Learning Approach for Morphological Analysis of Tigrigna Language Verbs

By

Gebrearegay Kalayu Abraha

APROVAL BY BOARD EXAMINERS

Dr. Yalemzewd Negash	_____
Dean, the School of Electrical and computer engineering	Signature
Dr.Eng. Getachew Alemu	_____
advisor	Signature
_____	_____
External Examiner	Signature
_____	_____
Internal Examiner	Signature

Acknowledgment

I would like to thank God for giving me strength to stay tuned throughout my goals and to complete this thesis work.

I would like to express my deepest gratitude to my advisor, Dr.Eng Getachew Alemu, for his continuous follow up and supervision throughout the whole work of this thesis and all rounded advices that he offered me. I really appreciate him for his constructive comments and for making me be the dreamer of my bright future. Every time he met me, he had been encouraging and offering me a long term dream in addition to the supervision of this thesis work.

I would also like to thank to Mr. Menore Tekeba for his decisive , constructive comments and suggestions that he gave me starting from the proposal. I really appreciate him for his professional and encouraging comments and suggestions.

My deepest gratitude to my family who had been supporting, encouraging me since my start of schooling.

My Gratitude to all my instructors, my friends and classmates, and the community of AAiT in general

Table of Contents

List of Tables

List of Figures

List of Acronyms

GB: Gigabit

IB1: Instance-Based1

IB1-IG: Instance Based1-Information Gain

ID: Inverse Distance

IG: Information Gain

MBL: Memory-Based Learning

NL : Natural Language

NLP : Natural Language Processing

NLU: Natural Language Understanding

RAM: Random Access Memory

TiMBL: Tilburg Memory-Based Learner

TLM : Two-Level Morphology

List of Symbols

A :Third person singular feminine subject marker

B : Third person singular masculine subject marker

C: Third person plural feminine subject marker

D : Second person plural feminine subject marker

E: First person singular object marker

F : Second person singular feminine subject marker

G : First person plural Object marker

H : Second person singular feminine object marker

I : First person Singular subject marker

J : Second person singular masculine object marker

K : second person plural feminine object marker

L : Second person plural masculine object marker

M : Second person singular masculine object marker

N : Second person plural Masculine subject marker

O : Third person singular feminine object marker

P : Passive Prefix

Q : Third person singular masculine object marker

R : Third person plural masculine subject marker

S : Stem

T : Third person plural feminine object marker

U : Third person plural masculine object marker

W : First person plural subject marker

Y : causative prefix

Z : negative prefix

0 : indicates Nothing(no boundary)

Abstract

Morphology, in linguistics, is the study of the forms of words that deals with the internal structure of words and word formation. Morphological analysis is the basic task of natural language processing that is defined as the process of segmenting words into morphemes and analyzing the word formation. It is often an initial step for various types of text analysis of any languages. Rule-based approach and machine learning approach are basic mechanisms for morphological analysis. The rule-based method is popular for the analysis but has limitations in terms of the efforts needed and the time. This is because the languages have many rules for a single word especially in the case of verbs. It is also difficult to include all words that need independent rules which limits the rule-based approach to accommodate words that are not in the database of the systems which can also affect the efficiency of the systems.

In this work, a system for morphological analysis of Tigrigna language verbs is designed and implemented using machine learning approach. It is intended to automatically segment a given input verb into morphemes and give their categories based on prefix-stem-suffix segmentation. It gives the inflectional categories based on the subject and object markers of verbs that includes the gender, number and person by detecting the correct boundary of the morphemes. The negative, causative and passive prefixes are also considered. The data needed for training and testing was collected from scratch and annotated manually as the language is under-resourced. After the annotation process, an automatic method was implemented using java to preprocess the annotated verbs to produce list of instances for training and testing. The instance- based algorithm was used with the overlap metric with information gain weighting (IB1-IG) and without weighting (IB1) the features.

Experiments were performed by varying the number of nearest neighbors starting from one up to seventeen where the accuracies were almost saturated for both the IB1 and IB1-IG. The majority class voting and the inverse distance weighted decision methods were also compared in the experiment. The best performance were obtained with IB1 using both decision methods when the number of nearest neighbors parameter was smaller. The performance decreased as the number of nearest neighbor increased for both decision methods but showed higher variation in the case of majority class voting. Similarly, the performance with IB1-IG was also better for the smaller number of nearest neighbor for both decision methods and decreased when the number of nearest neighbor increased where it showed higher decrement in the case of majority voting. The IB1 achieved better performance compared to the IB1-IG. A highest accuracy of 91.56% and 89.15% was achieved using IB1 and IB1-IG, respectively with the number of nearest neighbor parameter of 1 for IB1 and 2 for IB1-IG. This encouraging result revealed that the instance-based algorithm is able to automate the morphological analysis of Tigrigna verbs.

Keywords: Morphological analysis, Tigrigna verbs, data annotation, Instance-based, Accuracy

CHAPTER 1

Introduction

1.1. Background

Natural Language Processing (NLP) is an area combined from Artificial Intelligence and Linguistics that is intended to make computers understand the statements or words written in human languages. This came into existence to ease the user's work and to satisfy the wish to communicate with the computer in natural language. Since all users may not be well-versed in machine specific language, NLP caters those users who do not have enough time to learn new languages or get perfection in it. Natural Language Processing basically can be classified into two parts i.e. Natural Language Understanding and Natural Language Generation which evolves the task to understand and generate the text. It is an interdisciplinary field which aims at getting computers perform useful tasks involving natural language such as enabling human-machine communication, improving human-human communication or simply doing useful processing of text and speech[1]. Processing the morphology of natural languages is one of the basic task of natural language processing.

Morphology, in linguistics, is the study of the forms of words, and the ways in which words are related to other words of the same language. It deals with the internal structure of words and word formation, including affixation behavior and pattern properties. Formal differences among words serve a variety of purposes, from the creation of new lexical items to the indication of grammatical structure. It is concerned with the form and lexical meaning of words. The form corresponds to the orthographic representation with respect to written or printed symbols and the meaning to a set of related words which share a morpheme [2]. It is the study of the structure of a word and how it is built. It is described as small building blocks of a word that are called morphemes and they can be divided into two groups, stems and affixes. A stem is the main part of a word and affixes are morphemes that are added to a stem to give different meanings to it. A morpheme is the smallest orthographic unit that bears a meaning. There are two types of morphemes: free and bound morphemes depending on whether they can occur on their own as independent words or not. Bound morphemes can be further subdivided into roots, stems and affixes. In the word dogs for example, dog is the stem and -s is the affix. Using affixes allow a word to occur in different forms, it can give the different inflections and derivations. How common these variations are differs between different languages [3]. As explained in [4], there are many ways to combine morphemes to create words. Four of these methods are common and play important roles in speech and language processing: inflection, derivation, compounding and cliticization. Inflection is the combination of a word stem with a grammatical morpheme, usually resulting in a word of the same class as the original stem, and usually filling some syntactic function like agreement. For example, English has the inflectional morpheme *-s* for

marking plural nouns, and the inflectional morpheme *-ed* for marking the past tense on verbs. Derivation is the combination of a word stem with a grammatical morpheme, usually resulting in a word of a different class, often with a meaning hard to predict exactly. For example the verb computerize can take the derivational suffix*ation* to produce the noun computerization. Compounding is the combination of multiple word stems together. For example the noun doghouse is the concatenation of the morpheme *dog* with the morpheme *house*. Finally, cliticization is the combination of *Clitic,* a word stem with a clitic. A clitic is a morpheme that acts syntactically like a word, but is reduced in form and attached (phonologically and sometimes orthographically). For example, the word 'have' can be cliticized into 've.

According to Hedlund et al. [3], a language can be considered to be simple or complex with regards to morphology. English is considered to be simple while Swedish is a language that is considered to be morphologically complex. Tigrigna is one of these morphologically complex languages that is taken specifically to this study. For example a past form of a verb has independent inflections for all persons[24].

Morphological analysis is the basic task of natural language processing that is defined as the process of segmenting words into morphemes and analyzing the word formation. It is often an initial step for various types of text analysis of any languages [4]. This analysis and other applications of natural language processing can be achieved through different mechanisms, particularly a kind of rule-based mechanism and an automated mechanism through machines. In the case of the rule-based one, strict rules are created for the specific application or process according to the rules of the specific language [14]. These rules can be achieved either using a manual process by experts of that specific language or using set of programs organized as a software that processes the given task based on the instructions given to it. In the case of the automated mechanism, rules of the language for the specific application are learned through the use of large similar data which is known as the training data set. This is done using a machine learning technique, set of mathematical algorithms, which is currently being used for many different applications. This technique generally helps to minimize the effort taken by humans for large tiresome tasks and to improve the efficiency that can be taken as limitations for the rule based one through learned automation.

1.1.1. Machine Learning

Machine learning is one of the methods of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that machines should be able to learn and adapt through experience. It gives computers the ability to learn without being explicitly programmed. Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM[6]. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms can overcome the problem of following strictly static

programs through data-driven predictions or decisions, by building a model from sample similar data. Machine learning is applied in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible. It is also highly related to computational statistics, which also focuses on prediction-making through the use of computers. It has strong relation with mathematical optimizations, which gives methods, theory and application domains to the field [6]. Machine learning (in the context of text analytics) is a set of statistical methods for identifying some aspect of text such as parts of speech, entities, sentiment, morpheme etc. The methods can be expressed as a model that is then applied to other text that are unseen, or it could be a set of algorithms that work with large sets of data to analyze meaning [5]. Machine learning tasks are typically divided into two broad categories that bases on whether there is a learning signal available to a learning system. These broad categories are known as supervised and unsupervised [6].

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In this type of learning, each example is a pair consisting of an input object and a desired output value. A supervised learning algorithm analyzes the training data and outputs an inferred function, which can be used for generalizing new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a reasonable way [7].

Unsupervised machine learning on the other hand, is the machine learning task of inferring a function to describe hidden patterns from unlabeled data. Since the examples given to the learner are unlabeled, there is a difficulty of evaluating the accuracy of the structure that is output by the relevant algorithm which is one way of distinguishing unsupervised learning from supervised learning [2]. The goal is to find the regularities in the unlabeled input. There is a structure to the input space such that certain patterns occur more often than others, and it is needed to be seen what generally happens and what does not. The final goal of unsupervised learning is to group data into clusters. Generally, the basic task of unsupervised learning is to develop classification labels in an automatic way. The algorithms used in this type of learning are to seek out similarity among pieces of data to determine whether they can form different groups or clusters [7].

1.2 Statement of the Problem

Tigrigna is a member of the Ethio-Semitic languages, which belong to Afro-Asiatic super family [18]. This language is spoken primarily in Eritrea and Ethiopia having about 8.5 million speakers in these regions[1]. This indicates that the language has considerable native users which can make researchers work on different areas of natural language processing including morphological analysis. Language resources are important for those working on computational methods to analyze and study languages. However, to best of our knowledge, Tigrigna is under-resourced.

[1] http://www.nalrc.indiana.edu, [Accessed Dec 13,2017]

Many researches were done on morphological analysis for different languages. Two main methods of solving have been used in these areas, rule-based and corpus (machine learning) based [14]. The rule-based method is popular for different natural language processing applications including the morphological analysis; however, this method has many limitations as it is explained in [8]. Some of the limitations of this method are that it is time consuming, it is tedious task and expensive in terms of cost. Mehadi Yonis [9] also explained that the failure of rule-based morphological analyzer in his work was on the unknown words that were not in the database and the unknown rules that were not set. This indicates that systems developed using this method have effects on their performances. The rule-based method is affected with the language complexity [20]. The process of this analysis may seem to be fairly easy for the languages that have small inflectional properties though the task is huge. English is one of these languages that can be easily taken as language with fewer inflections [3]. However, it is not easy for languages that have high inflections. Tigrigna is one of the complex languages in its inflectional property where a single word can have many inflected forms that have different meanings based on the patterns of the affix attached to that word [24]. This inflectional behavior is dominant in Tigrigna verbs that make the rule-based method difficult for its analysis. The inflectional difficulty also contributes in addition to the common limitations of the rule-based systems. Having these general limitations of the rule-based method, the second method, the machine learning based, is on use for many languages as an improving method as it is explained in [10]. However, to the best of my knowledge, the machine learning based method was not used for Tigrigna language which made us propose for this study.

Many researchers had worked on similar area for different languages as explained in Chapter Two such as Gasser and Wondwosen [11] for Amharic verbs. However, the language rules are different for different languages which are learned by the algorithms differently. In summary, this work is proposed considering the limitations of the rule-based method and the resource scarcity of Tigrigna verbs. The following research question is set to be answered.

Research Question: can the instance-based algorithm be applied for morphological analysis of Tigrigna verbs?

1.3. Significances of the Research

One of the challenging tasks in machine learning related research work is the preparation of the data for training and testing of a model to be built. In the current work, the raw data (verbs) were collected manually and the annotation process was also done manually. Having done this, 4,982 raw verbs were collected and 5,744 list of instances prepared from 718 annotated verbs were prepared for direct use for training and testing purpose. This data will help to add a research resource for the area which will help to extend the current work and for other researchers who need the same data. A preprocessing method is developed which will help to produce additional instances for training and testing. To the best of my knowledge, the morphological analysis of the language including the verb was not done using the machine learning technique so far which

indicates that the current work is crucial that is delivered as local research result. It will open a way to further study towards the language on the analysis of its morphology in general and the related applications of the language using machine learning techniques.

The results of a morphological study are essential from the point of almost all practical applications that deal with natural language processing. After all, an application must first recognize the word in question before analyzing it syntactically, semantically, or whatever the level may be. The morphological strength of Languages demands the use of thorough morphological analysis. Morphological analysis should be the first step towards any language processing end user applications [12].

Translation of two languages with highly different morphological structures as explained by Youn-gsuk lee [12] in the case of Arabic and English poses a challenge to successful implementation of statistical machine translation models. Rarely occurring inflected forms of a stem in Arabic often do not accurately translate due to the frequency imbalance with the corresponding translation word in English. So called a word (separated by a white space) in Arabic often corresponds to more than one independent word in English, posing a technical problem to the source channel models. This indicates, therefore, the morphological analysis of languages in general is very crucial to the development of efficient machine translation application among different natural languages. Having an efficient model for morphological analysis for Tigrigna therefore helps for the development of an efficient machine translation indirectly.

Morphological analysis is also applied in vocabulary production for users of one's language. According to Toms S. Bellomo [13], building a vocabulary strategy program based on morphological analysis that includes word parts that are stable in form and transparent in meaning will not be of much use if these parts assist in recalling or learning only a few words. Ideally, selected morphemes should transfer to multiple words that will allow the student to obtain much mileage from this strategy. This is helpful for Tigrigna language learners too.

1.4. Objectives of the Study

1.4.1 General Objective

The general objective of this work is to design and implement a model for morphological analysis of Tigrigna verbs through machine learning technique.

1.4.2 Specific Objectives

The specific objectives are:

- To collect and prepare annotated Tigrigna verbs text corpus

- To develop or adopt an algorithm for preprocessing of the annotated data
- To design a model and analyze it to extract its performance

1.5. Scope of the Study

The morphology of a natural language generally describes the words of that language which includes all parts of speech of the language. Verb is one of the most dominant parts of speech of most natural languages which can have many forms for a single stem verb. The study done in this work is limited to the morphology of Tigrigna verbs as this part of speech is the most variant and dominant of all the other part of speech of that language. Therefore, the scope of the study in the case of the morphology is only for Tigrigna language verbs. In the case of the morphological analysis of natural languages, there are many tasks that can be taken under the analysis process. Part of speech tagging is one of these tasks of the analysis where the goal is to indicate what part of speech a given word is. Another task is the analysis of the phonemes of a given word of the natural language. The morphological analysis for the verbs is therefore the proposed work that mainly focuses on the segmentation and analysis on the segmented morphemes of verbs through boundary detection using supervised machine learning technique.

1.6. Organization of the Thesis

The next part of the thesis is organized in five chapters. Chapter Two describes the literature part that mainly consists different research works that are related to the proposed task. It introduces the general concepts of the related areas and its importance. It discusses the general approaches that are used for morphological analysis including some research works done based on these approaches. Finally, it summarizes the whole point and declares the importance of the current work.

Chapter Three is all about the morphology of Tigrigna language. It describes the language in general as well as the word formation of the language. It describes the inflectional and derivational morphology of the verbs, nouns and adjectives of the language.

In Chapter Four, the main methodology is described including the data collection, the model architecture and its description. Each task in the model architecture is described in detail. Chapter five is all about the result and its discussion. It describes the system in general and consists the simulation results. A comparison of different results is described which finally summarizes the best results. Chapter Six concludes the whole work point with further recommendations. The final section consists list of references followed by the appendixes.

CHAPTER 2

Literature Review

2.1 Introduction

Allen [14] stated that most of human knowledge is recorded in linguistic form, i.e., in the form of natural language (NL) texts and utterances. This reliance on NL makes understanding of natural language crucial for improved knowledge representation. Since the invention of computers, there are also efforts to develop computer system that understand natural languages. Such systems are referred to as Natural Language Understanding (NLU) systems. Allen indicated that NLU systems can be developed at different level (such as phoneme, word and sentence levels) and integrated to form a full-fledged Natural Language Processing (NLP) system.

NLU systems at sound (phone) level are used to identify the phonological features of phones used in a language. At word level, NLU systems are developed to understand words of a language, i.e., to understand what constitutes words (morphemes), how morphemes combine to form words and what morphemic components of a word contribute to the overall meaning of the word. Other system such as part of speech taggers and sentence parsers are developed for higher-level linguistic processing such as syntactic and semantic analysis [14].

Systems at word level, called morphological systems, are required because of the fact that knowledge of words of a language can't be summarized in a finite list. That is, words can be derived, conjugated, and used in a number of ways. For example, from the word play, we can generate many other words like 'playing', player', 'plays', 'players', 'base-ball players' and so on. However, it is technically and practically difficult to prepare exhaustive list of words of a language for such applications like dictionary compilation. The preferred method is to know different patterns (principle) of word-formation of the language and apply them for the required applications. It is for this reason that computerized morphological systems are developed. They are used to enable computers understand words based on the principle of word-formation of a particular language[15].

2.2 Approaches for Morphological Analysis

There are a number of approaches employed in computational morphology. As discussed in [16], some of these approaches are based on concepts in automata theory, probability, principle of analogy, and information theory. These are broadly categorized into rule-based and corpus-based(machine learning) approaches. A rule-based approach is based on a theory of morphology laid down by an expert. This approach enables to incorporate sophisticated linguistic theories such as generative phonology into computational morphology processes. Because of their

reliance on linguistic theories, systems developed using rule-based approaches are often efficient and produce better quality outputs when the rules are correct and simple(small).

The most commonly cited rule-based method is the Two-Level-morphology (TLM) [17]. TLM is devised to handle morphological analysis and generation in a bi-directional way. The approach is based on two lexicons (one for the underlying and the other for surface word forms), and a set of morphological rules. The rules establish whether a given sequence of characters at the surface level (as it appears in the text) can correspond to a sequence of symbols used to represent the morphemes in the lexicon. In other word, the rules map the two strings to each other. TLM is currently very popular method in computational morphology.

Unlike rule-based approaches, corpus-based approaches do not strictly follow explicit theory of linguistics. These approaches use some algorithms to learn, say about the morphological segmentation of a language, from an input data (corpus). The knowledge acquired is then used to perform the morphological analysis task [10].

Based on the type of text corpora used, corpus-based approaches can be further categorized into supervised and unsupervised approaches as described in the background section. Supervised approaches use annotated text corpora while unsupervised approaches uses natural corpus as those found in newspaper and books. Annotated text can be word-forms tokenized into constituent morpheme by human expert or words with their grammatical properties assigned pre-hand. The work of Nagamatsu and Tanaka on Japanese morphology is an example. They used morphologically annotated dictionary to train their k-NN-based system. The system employs n-gram data to determine the best point of morphological segmentation of sentences. The unsupervised approaches, on the other hand, do not need such preprocessing on the corpora[10].

Many researches had been done on the area of morphological analysis for many different languages to produce natural language processing applications. These researches are categorized into the two mechanisms of the ways the researchers followed, as explained above, rule-based and machine learning based approaches. The following sections describe both the approaches used for the analysis of different languages.

2.2.1 Rule-based Approach

In the case of the rule-based approach, different researchers worked on the development of morphological analysis for different languages. One of these is a research done by Mahdi Yonis in 2017 for Af-somali language[9]. In that paper, the main objective of the work was to develop morphological analyzer for the language. The author described that an accuracy of about 87% was achieved by using a finite state transducer, a kind of rule-based technique for natural language processing applications, in which a database of the rules of morph-tactics is used. This technique analyzes good for the available rules and words fed to the system. It includes all types of parts of speeches for the language which has a better domain of the analysis. The author explained that the failures of the system was more on complex words and words whose stems

are not stored in the database of the rules. As the system uses the guesser for the unknown words, it results in failing in identifying the correct analysis of given words as an input to the system. This indicates that the system cannot accommodate all the words to be analyzed as it is hard to collect all rules and vocabularies of the language. This is one of the limitations of most of the rule-based techniques. Therefore, the unknown words that are not available in the set of the rules are becoming out of the correct analysis which leads the system to be inefficient when it is given new words. Though the result achieved by this system is fine for the available rules and collected vocabularies, it needs to be improved to analyze the unknown sets by using a mechanism that generalizes based on the available sets which is the target method to be used for Tigrigna language in this work. This mechanism improves the limitations of rule-based methods in general.

Another work related to some extent to this work is a development of stemming algorithm done by Yonas Fissaha[18]. The main objective of this work was to devise a rule-based algorithm that extracts the stem of a given Tigrigna language text. In this work, the author tried to set rules by taking a list of the possible affixes of the language using a conditional structure for each affix. The author set many rules for each of the types of suffixes and prefixes separately. The aims of these rules are to remove the affixes to extract the stem of the input word. The similarity of this work with the proposed one is on giving the stem of the given input word when the word is a verb. However, the proposed one does not remove the affixes, it only shows the formation of the input verb. The author tried to list some of the affixes of the language while it has plenty of affixes that can be used within a single word. When it is compared on the similarity, the stemming of the author cannot handle all the affixes of the language as it will fail for those which are missing from the list. Having that limitation, it is important to improve these rule-based algorithms that could not handle all the prefixes and suffixes of the language in order to improve the efficiency. The system was measured with the counting of the correct stems found from the collected data achieving about 85% of accuracy. The author stated that the error percentage results were also taken as measuring mechanism which is about 15%. The author described that the causes of errors were mostly from the over stemming problems which accounts about 77%. These problems can be improved by using a machine learning approach that learns the general patterns of the affixes that enables to predict for the new affixes that are not included in the rules.

Another work done by Ramchandra P. Bhavsar and B. V. Pawar[19] was a rule-based framework that has rule format that is flexible to include rules for different languages. The objective of this work was to develop a rule driven framework in which morphology rules are stored in database table in the generic format such that rules of different languages should be accommodated seamlessly. After the rules are created, appropriate rules are applied to stem word depending on its lexical category. The authors described that the rules may or may not be applicable to all words in that lexical category but is applicable to only subset of words with some common traits or subclasses under that lexical category. The framework can be integrated with lexicon creation tool to generate inflectional as well as derivational word morphology. It

automates the process of word generation which minimizes the time and effort needed. It is also user friendly and flexible as it uses a wild card notation. It does not have font related dependencies as the data is stored in a Unicode format. These features make the framework be nice for the relevant application but are limited as it does not accommodate rules that are not in the database that leads to lower efficiency. Though the proposed work is for the verbs only, it learns the general patterns (by storing) of the language rules as it uses machine learning approach which can predict for the analysis of unseen verbs.

Gasser [20] has also worked on development of a system for morphological processing of three languages; Amharic, Tigrigna and Afan-oromo. The goal of this work was to develop a kind of software for analyzing and generating the word forms of the three languages considering the languages have complex morphology and have no software for this task. The analysis task of the software is to break down a given input nouns and verbs into morphemes and give their grammatical categories. The author used a weighted finite state transducer, an approach of which each of the arcs in a transducer is weighted with a feature structure. As the arcs in the finite state transducer are traversed, a set of feature-value pairs is accumulated by unifying the current set with whatever appears on the arcs along the path through the transducer. The result of the traversing during the analysis outputs character sequences of the root and the grammatical categories. The system works for the adjectives, verbs and nouns of Amharic and Afan-oromo while it works only for verbs in the case of Tigrigna having different functions for each of the languages. The author stated that the system was evaluated using manual observation by looking at dictionaries. He stated that there was a limited resource for the case of Tigrigna compared to the other languages. Only 602 root verbs were used for Tigrigna which is very small. The accuracy was measured by taking 200 random inputs from dictionary achieving 96%. The system achieved 99% for Amharic verbs, 95.5% for Amharic adjective and nouns.

2.2.2 Machine Learning Approach

In the case of machine learning approach, the rules of a language is not set manually or are not strict rules to be followed. The rules are learned by using data set during training which tries to generalize the patterns of the given language so that it knows without much errors when new data is input to it. This can be either by collecting large unlabeled data and use it for training using unsupervised learning mechanism or supervised mechanism where pre-annotated data set is provided for the learning process. The following papers are some of the areas where machine learning approach is used for analyzing the morphology of different languages.

J. Gold Smith [21] described an algorithm used for the unsupervised learning of natural language morphology which works well for European languages and other languages in which the average number of morphemes per word is not too high. The main focus of this work was on morphological analysis based purely on distributional information, and in particular on the task of segmenting a word into distinct, successive morphs rather than the assignment of morpho-syntactic features, for example, which is the goal of many other morphological parsers under

development today. The main goal was in the consideration that good morphological parsers for many of the world's languages would be useful for a number of functions, ranging from document retrieval to automatic machine translation, all of which would arguably be superior if trained from a corpus in which words were morphologically segmented. The minimum description length method, a model that defines a description length of the corpus given a probabilistic model of the corpus, was used. Accuracy was used to evaluate the performance of the model by preparing a gold standard dataset. Linguistica learning tool was used to simulate the implementation. An accuracy of 72% was achieved using this method. As the author described that the method works well for European languages and languages that have small number of morphemes per word, this limits it that the complex languages and non-European languages are to be out of the advantage of this result.

Wondwosen and Gasser worked on morphological analysis for Amharic verbs using supervised machine learning. They used Inductive Logic Programming (ILP), implemented in CLOG. CLOG learns rules as a first order predicate decision list. The main objective of this paper was to get a machine learned stem rules extractor from a give data. They prepared the data manually in the way the predicate structure needed by the learning process. They achieved about 87% accuracy of the analysis. In this paper, the subject prefixes and suffixes only were addressed in the analysis of the Amharic verbs [11]. However, there are plenty of object suffixes in languages in general and in Amharic language as well. Though Tigrigna and Amharic are both Semitic languages and have similarity, the rules of the language in general have many differences including the suffixation of the verbs in terms of the object suffixes which is not handled in this work.

Another paper was done by Xuri TANG [22] for English morphological analysis using machine learned rules where morphological rule learning and morphological analysis components are included in the study. In this paper, the main objective was to understand the inner mechanism of word form formation of English language. The author adopted an approach for learning affix rules from word list and tested it using word list of different scales and achieved an average result of 81%. The author explained that a language specific rule considerations have improved the result for English compared to the other previous algorithms used for similar languages. This indicates that the morphological rule of languages should be considered to obtain a better result of language models. The results achieved are good when it is considered for the language modeled, particularly, English. However, the morphological rules of English and Tigrigna language are totally different which indicates that Tigrigna needs its own rule learning mechanism that must be designed using state-of-the-art algorithms. The difference here is the rules and internal structure of the language morphemes which should be considered independently.

2.4 Summary

Natural languages have crucial advantages in the representations of human knowledge as it is recorded in a linguistic form. This requires systems that can understand the natural languages which can be developed at different levels such as the sentence level and word level that are integrated to form a full-fledged natural language processing as it is explained in[14] . Systems at word level, called morphological systems, are required because of the fact that knowledge of words of a language can't be summarized in a finite list. In other words, words can be derived, conjugated, and used in a number of ways. Many researchers have been doing their study on the area of natural language processing particularly, the morphological analysis of different languages which is the intermediate task of natural language processing applications. They tried to use different mechanisms for the analysis of the morphology of the languages which are categorized as rule-based and machine learning approaches as explained in the above sections. The category is based on that either the techniques use a strict rule to be followed or a sample for generalizations. The rule-based technique uses the first and the machine learning technique uses the latter in the cases categorized above. The rule-based one results in good performances for what is given to it provided that the rule is perfect and simple. However, it is very limited; time consuming, expensive and tiresome. This becomes more complex for languages that have complex morphologies such as Amharic and Tigrigna. Each natural language has its own morphological patterns irrespective of the characters they use. This is the main point that different researchers are doing a model for different natural languages as they have their own morphological rules. This indicates that each language needs to have its own model. Tigrigna language has many morphemes for a given word. The most variant part of speech that can have many forms is the verb. A single verb can be written in many forms by attaching the affixes. The rule-based technique is tried on extracting stems of a given surface verb as explained above which is part of the morphological analysis. But this technique is so limited that cannot accommodate all verbs as stated by the authors. On the above literatures, both the machine learning approach and the rule-based approach have been used for the task for different languages. The machine learning (the supervised one) is selected for the proposed study as it is better for the morphologically complex languages. This is because the method generalizes the rules through training.

Tigrigna language is one of the under-resourced languages that is lagging behind in the world of computational tasks researches, particularly the automated one. Since the models designed for the other languages cannot do well for Tigrigna for the reason they have different morphological rules and complexity, it is important to design its own model by using or adopting the state-of-the art algorithms. To the best of the literatures reviewed, Tigrigna language has no corpus of its verb morphology and is therefore important to prepare a corpus of Tigrigna verbs which will be used for further study on similar areas. Therefore, doing a research study on this area using machine learning technique is so crucial for the language which will also be enhanced well for further study.

CHAPTER 3

Morphology of Tigrigna Language

3.1 Introduction

Morphology is the branch of linguistics that deals with the internal structure of words and word formation, including affixation behavior, roots, and pattern properties [23]. Morphology is the main source of variation in natural language text, with suffixing and prefixing being the most common ways of creating a word variant. Morphology can be classified as either inflectional or derivational. Inflection is variation or change of form that words undergo to mark distinctions of case, gender, number, tense, person, and comparison. Inflectional morphology is applied to a given stem with predictable formation. It does not affect the word's grammatical category, such as noun, verb, etc. Case, gender, number, tense, person are some examples of characteristics that might be affected by inflection. Derivational morphology, on the other hand, concatenates to a given word a set of morphemes that may affect the grammatical and syntactic category of the word.

A word can have several word forms, e.g., the word "write" can take the forms "writes", "wrote" and "written", usually called inflected forms. The root is the original form of the word before any transformation process, and it plays an important role in language studies. The root is the form of a word from which the other forms can be derived using the morphological rules of a language. A morpheme is the smallest unit of a language that has a meaning and cannot be broken down further into meaningful or recognizable parts and should impart a function or a meaning to the word which they are part of. An affix is a morpheme that can be added before (prefix) or after (suffix), or inserted inside (infix) a root or a stem to form new words or meanings. Morphological information of a language is useful for several natural language applications such as stemming, morphological analysis, text generation, machine translation, document retrieval [18].

3.2 Overview of Tigrigna Language

Tigrigna is a member of the Ethio-Semitic languages, which belong to Afro-Asiatic super family [18]. Tigrigna is spoken primarily in Eritrea and Ethiopia. According to the literatures reviewed, there are about 8.5 million Tigrigna speakers in these regions. Tigrigna is written in the Ge'ez script which is originally developed for Ge'ez language. In Tigrigna, each symbol represents a consonant and vowel combination and the symbols are organized in groups of similar symbols on the basis of both the consonant and the vowel. For each consonant in each symbol, there is an unmarked symbol representing that consonant followed by a canonical or inherent vowel. Tigrigna like other Semitic languages such as Arabic and Amharic exhibits a root pattern

morphological phenomenon. In addition, it uses different affixes to create inflectional and derivational word forms [18].

3.3. Tigrigna Morphological System and Word Formation

In Tigrigna language, each word group generates an increased verb forms and noun forms by the addition of derivational and inflectional affixes. Words in Tigrigna are built from the roots by means of a variety of morphological operations such as compounding, affixation, and reduplication. An affix in Tigrigna is a morpheme that can be added before or after, or inserted inside, a root or a stem as a prefix, suffix or infix, respectively, to form new words or meanings. Tigrigna affixes have the feature of concatenating with each other in predefined linguistic rules. This feature increases the overall number of affixes [18]. There are also some prefixes and suffixes which determine whether a word is a subject marker, pronoun, preposition, object marker or a definite article. Tigrigna is highly productive, both derivationally and inflectionally. Definite articles, conjunctions, articles and other prefixes can attach to the beginning of a word, and large numbers of suffixes can attach to the end. A given headword can be found in huge number of different forms. Tigrigna concatinative morphology regulates how a stem and affixes glue together, while non-concatenative one combines morphemes in more complex ways. Affixes in Tigrigna can be classified as four categories. Prefixes precede the base form, such as *`ዘይ-,`ኣይ-, ተ-, ስለ-,* etc are some of the prefixes that can be attached to different words. Suffixes follow the base form. For instance, -ኩም, -ክን,-ት, -ታት etc. are some form of suffixes that can be attached at the end of many words. Finally, Infixes are inside the base form. Circumfixes are affixes attached before and after the base form at the same time. While circumfixes formally are combination of allowed prefixes and suffixes, they have to be treated as discontinuous units for semantic and grammatical reasons. Tigrigna non-concatenative morphology refers to reduplicated morpheme forms. Reduplicated words based on morpheme regularity are grouped into full reduplication (e.g., the word ስትይስትይ is derived from the root ስትይ) and partial reduplication of different kinds. The latter includes reduplicated stems with affixes (e.g. word ሰባበረ is derived from stem ሰበረ , ተረጋገሙ is derived from stem ረገመ , the word ፈገምገም is derived from the stem ፈገመ) and there are also various irregular reduplications[18].

3.4. Derivational and Inflectional Morphology

There are five parts of speech in Tigrigna: adjectives, nouns, verbs, adverbs and prepositions[18]. Prepositions and conjunctions are totally unproductive. Adverbs are few in number and are less productive. Therefore, the discussion of derivational and inflectional morphology concentrates on the remaining three parts of speech, namely verbs, nouns and adjectives.

3.5. Inflectional Morphology of Tigrigna

As Tigrigna is a highly inflectional language, definite articles, conjunctions and other prefixes can be attached to the beginning of a word, and large numbers of suffixes can be attached to the end. A given root of word can be found in huge number of different forms.

3.5.1. Inflection of Verbs

A verb is one of the parts of speech in languages that express actions and situations. A sentence cannot stand without a verb to give a complete meaning. In Tigrigna, the words under this category are put at the end of a sentence [24]. A significantly large part of the vocabulary consists of verbs, which exhibit different morpho-syntactic properties based on the arrangement of the consonant-vowel patterns. For example, the root ሰብር, meaning 'to break' can have the perfect form ሰበረ, with the pattern CVCVCV(C-consenant, V-vowel), imperfect form ትሰብር with the pattern CCVCC, gerund form ሰቢርካ with the pattern CVCVCCV, imperative form ስበር, with the pattern CCVC, causative form ኣስበረ with the pattern as-CVCV, passive form ተሰበረ with the pattern CVCVCV. Subject, gender and number are also indicated as bound morphemes on the verb as well as object markers and tense producing complex verb morphology [18].

The simplest form of the verb is the third person masculine singular of the perfect tense. In most Tigrigna dictionaries, all the words derived from a tri-literal root are entered under the third person masculine singular form of the verb. Each three-consonant (or "tri-literal") root belongs to one of three conjugation classes conventionally known as A, B and C. This division is a basic feature of Ethiopian Semitic languages. Most three-consonant roots are in the A class. These categories have no germination but the vowel 'ኣ' appears between both pairs of consonants that is hidden. Examples are: ደረፈ meaning, "he sung", ደየበ meaning,"he climbed", ሰተየ meaning,"he drank". The B class is distinguished by the gemination of the second consonant in all forms. Some examples are: ደቀሰ meaning, "he slept", ወሰኸ meaning,"he added". The relatively few members of the C class take the vowel 'ኣ' between the first and second consonants. Examples are ባረኸ meaning, "he blessed" and ናፈቐ meaning,"long for, miss". Tigrigna also has a significant number of four-consonant (or "quadriliteral") roots. These fall into a single conjugation class. Examples are መስከረ meaning,"he testify" and ቀልጠፈ meaning,"he hurried" . The language also has five-consonant (or "quinquiliteral") roots. Most, if not all, of these are "defective" in the sense that their simplest form takes the 'ተ-' prefix. Examples are ተንቀጥቀጠ, meaning, "it trembled" and ተምበርከኸ meaning, "he kneeled".

Tigrigna verbs have two tenses: perfect and imperfect. Perfect tense denotes actions completed while imperfect denotes uncompleted actions. The imperfect tense has four moods: indicative, subjective, jussive and imperative. Tigrigna verbs in perfect tense consist of a stem and a subject marker. The subject marker indicates the person, gender and number of the subject. The form of a verb in perfect tense can have subject marker and pronoun suffix. The form of a subject-marker

is determined together by the person, gender, and number of the subject. Other elements like negative markers also inflect verbs in Tigrigna.

Each lexeme can appear in four different tense-aspect-mood categories, conventionally referred to as perfective, imperfective, jussive/imperative, and gerund. Every Tigrigna verb must agree with its subject. As in other Semitic languages, subject agreement is expressed by suffixes alone in some tense-aspect mood categories (perfective and gerundive) and by a combination of prefixes and suffixes. Tigrigna verbs also have a suffix representing the person, number and gender of a direct object or an indirect object that is definite.

Tigrigna verbs are inflected for person, gender, number and time with basic verb form being the third person masculine singular. In the verb ዘወረኒ meaning, "he circulated me", the stem is ዘወረ while the last morpheme, ኒ indicates the first person singular object. The verb ዘወረት meaning, "she circulated", has also two morphemes, ዘወረ and the last morpheme -ት. The last morpheme or -ት is the subject marker that indicates third person singular feminine. Tigrigna verbs have two tenses: perfect and imperfect. Perfect tense denotes actions completed, while imperfect denotes uncompleted actions. The imperfect tense has four moods: indicative, subjective, jussive, and imperative. Tigrigna verbs are conjugated in perfective, imperfective, indicative, subjective and imperative. In conjugating the verbs affixes are attached to the verbs.

Inflection of Perfective Tense: The perfect tense which is the basic form normally expresses the past tense and consist of a stem and a subject marker. The form of a verb in perfect tense can have subject marker and pronoun suffix. The subject marker indicates the person, gender, and number of the subject. The form of a subject-marker is determined together by the person, gender, and number of the subject. The following example demonstrates how suffixes are attached to verbs for indicating subject marker and pronoun.

Verb variations	**Person**			**Gender**		**Number**	
	1st	2nd	3rd	Male	Female	Singular	Plural
ቀቲለ	+			+	+	+	
ቀቲልና	+						+
ቀቲልካ		+		+		+	
ቀቲልኪ		+			+	+	
ቀቲልኩም		+		+	+		+
ቀቲልክን		+			+		+
ቀቲላ			+		+	+	
ቀቲለን			+		+		+
ቀቲሎም			+	+			+
ቀተለት			+		+	+	
ቀቲሉ			+	+		+	

Table 3. 1 : Inflection of perfective tense

Inflection of Imperfective Tense: The imperfect tense has four moods: indicative, subjective, jussive, and imperative and is inflected by prefixing and suffixing gender, person, and number morphemes to the imperfective verb stem. The table below shows how suffixes and prefixes are added for the verb "ገበረ".

Person	Singular	Plural
1st person	እገብር (`I-gebr)	ንገብር (n-gebr)
2 nd person-masculine	ትገብር (t-gebr)	ትገብሩ (t-gebr-u)
2nd person- feminine	ትገብሪ (t-gebr-i)	ትገብራ (t-gebr-a)
3rd person-masculine	ይገብር(y-gebr)	ይገብሩ (y-gebr-u)
3rd person-feminine	ትገብር (t-gebr)	ይገብራ (y-gebr-a)

Table 3. 2 : Inflection of imperfective tense

3.5.2 Inflection of Nouns

Tigrigna nouns inflect for case, number, definiteness, and gender. A noun has the nominative case when it is a subject; accusative when it is the object of a verb; and genitive when it is the object of a preposition. The form of Tigrigna noun is determined by its gender, number, and grammatical case. Most plural nouns are formed by adding a plural marker affix (-ታት,or ኣት,) to the singular form. Although when referring to groups belonging to a certain tribe or country is affixed. There are a set of affixes that are used to make plural nouns and are attached as prefix or suffixes to the nouns. The affixes -ታት, ኣት, ኣን, ኦት, ውቲ, ቲ are used as suffixes to inflect nouns. Here are some examples to show the inflection of nouns[18].

Noun	Noun-suffix	After suffixation
ባሕሪ	ባሕሪ-ታት	ባሕሪታት
እምባ	እምባ-ታት	እምባታት
ስእሊ	ስእሊ-ታት	ስእልታት
ሃገር	ሃገር-ኣት	ሃገራት
ሰማይ	ሰማይ-ኣት	ሰማያት
ምእመን	ምእመን-ኣን	ምእመናን
መምህር	መምህር-ኣን	መምህራን

Table 3. 3 :Inflections of nouns

Another form of inflection for nouns is created by attaching the morpheme ኣ as prefix. For example in the words ገረብ meaning, 'forest', ፈረስ meaning, 'horse', ከረን meaning, 'mountain', ኣግራብ meaning, 'forests', ኣፍራስ meaning , 'horses', ኣከራን meaning, 'mountains'.

Tigrigna has two grammatical genders: masculine and feminine, and all nouns belong to either one or the other and inanimate objects take one of the genders. Some noun pairs for people distinguish masculine and feminine by their endings, with the feminine signaled by ኢት and the masculine signaled by ኢ. These include agent nouns derived from verbs , ከፈተ meaning ,"he

opened", ከፋቲ meaning, 'opener (masculine)', ከፋቲት meaning , 'opener' (feminine)' and nouns for nationalities or natives of particular regions - ትግራዋይ meaning, 'Tigrean (masculine)', ትግራወይቲ meaning 'Tigrean (feminine)'.

3.5.3 Inflection of Adjectives

Tigrigna adjectives inflect for number and gender. Tigrigna adjectives may have separate masculine singular, feminine singular and plural forms, and adjectives usually agree in gender and number with the nouns they modify. The plural forms follow the same patterns as noun plurals; that is, they may be formed by suffixes or internal changes or a combination of the two. The affixes that are used for the inflections of the adjectives are ኡ, ቲ, ኣት, ኣን and ኦት. Table 3.4 shows inflection of some adjectives[18].

Singular adjectives	Plural adjectives	Affix attached
ቀያሕ	ቀያሕቲ	-ቲ
በላሕ	በላሕቲ	-ቲ
ሰነፍ	ሰነፋት	-ኣት
ኩቡር	ኩቡራት	-ኣት
ሓረስታይ	ሓረስቶት	-ኦት

Table 3. 4 : inflection of adjectives

3.6 Derivational Morphology

Derivational morphology describes how affixes combine with word stems to derive new words. Derivational affixes may affect the part-of-speech and meaning of a word.

3.6.1 Derivation of Verbs

Unlike the other word categories such as nouns and adjectives, the derivation of verbs from other parts of speech is not common. Almost all Tigrigna verbs are derived from root consonants as indicated in [24]. Traditionally, a distinction is made between simple and derived verbs. Simple verbs are those verbs derived from roots by intercalating vowel patterns whereas derived verbs are considered as derivatives of simple verbs. The derivation process can be an internal one in which consonant-vowel patterns are changed, an external one where derivational affixes are attached to the simple derived verbs or a combination of the internal and external derivational processes. The derivation of causative, passive, repetitive and reciprocal verbs are presented in the following paragraph.

Causative verbs are derived by adding the derivational morphemes 'ኣ-' to the verb stem as in the examples ደቀሰ meaning,"he slept" ,ኣደቀሰ meaning, "cause to sleep", በፅሐ meaning, "he arrived", ኣበፅሐ ,"cause to arrive" and ወሰደ meanin,"he took," ኣውሰደ meaning, "cause to take". In most cases the 'ኣ' morpheme is used to form causative of intransitive verbs, transitive ones and verbs of state. Some exceptions are the verbs that begin with 'ኣ', always take the morpheme 'ኣ' but add

the morpheme ‘እ’ after the morpheme ‘ኣ- to form causative e.g. ኣሰረ , ኣእሰረ. The passive verbs are derived using the derivational morpheme ተ- . This derivational morpheme is realized as ተ- before consonants and as ት- before vowels. Moreover, in the imperfect, jussive and in derived nominal like verbal noun, the derivational morpheme ት is used. In this case, it assimilates to the first consonant of the verb stem, and as a result, the first radical of the verb geminates. Some exceptions are intransitive verbs like ፈሊሑ meaning, “it is boiled”, that form their passive forms using the prefix ተ- as in ተፈሊሑ meaning, “it was boiled”. Such kind of verbs can derive their passive from their causative form. Reduplicative stems indicate an action which is performed repeatedly. For tri-radical verbs, such stems are formed by duplicating the second consonant of the root and using the ኣ- after the duplicated consonant as in ሰባበረ meaning, “he broke repeatedly” derived from the root ሰብር, ‘break’. All verb types, Type A, B and C have the same reduplicative forms. Reciprocal verbs are derived by prefixing the derivational morpheme ተ- either to the derived type C forms (that use the vowel, ‘ኣ’, after the first radical) or to the reduplicative stem. For example, reciprocal forms of ተቃተሉ meaning, “killed each other” and ተቀታተሉ meaning, “killed one another” are derived from the type C stem ቀተሉ and reduplicative stem ቀታተሉ , respectively[18].

3.6.2 Derivational of Nouns

Tigrigna nouns can be either primary or derived. They are derived if they are related in their root consonants and/or meaning to verbs, adjectives, or other nouns. Otherwise, they are primary. For example, a noun እግሪ meaning,’foot, leg’ is primary but, እግረኛ meaning, ’pedestrian’ ,is derived from the nominal base እግረ by adding the morpheme ኛ.

In Tigrigna, nouns can also be formed through compounding. For example, ቤትብልዒ meaning, ‘restuarant’, is derived from the nouns ቤት ,‘house’, and ብልዒ , ’food’.As it can be seen, no morpheme is used to bind the two nouns. But, there are also compound nouns whose components came together by inserting the compounding morpheme አ as in ቤተክርስትያን meaning, ’church’, which is formed from ቤት,’house’, and ክርስትያን meaning, ’Christian’.

3.6.3 Derivations of Adjectives

Adjectives in Tigrigna include all the words that modify nouns and can be modified by many ways. As it is true for nouns, adjectives can also be primary (such as ለዋህ meaning, ’kind’) or derived, although the number of primary adjectives is very small. Adjectives are derived from nouns, stems or verbal roots by adding a suffix and by intercalation. The suffixes ኣም, -ዊ , -ኣዊ , -ኣይ, -ኢታይ ,-ታይ , -አኛ and -ኣዋይ are used in the derivation of adjectives from nouns. For example it is possible to derive ሃፍታም meaning, ‘rich, wealthy’, ዘበናዊ meaning,’modern’, and ማእኸላይ meaning, ‘central’, from the nouns ሃፍቲ, ’wealth’, ዘበን, ’period’ and ማእኸል ,’center’, respectively. Adjectives can also be derived either from roots by intercalation of vocalic elements or attaching a suffix to bound stems[18].

CHAPTER 4

Methodology and Data Analysis

4.1 Data Collection

To the best of our knowledge, there is no Tigrigna verb corpus and no tagged words for use which made us collect the verbs manually. In the collection of the relevant data, different online resources are visited and collected as raw corpus from which the verbs are filtered. In this task, 4982 Tigrigna verbs are filtered (collected) that are used for annotations process where the annotated data is used as input for preprocessing task. Some of the main resources used are: VOA Tigrigna news[2], BBC Tigrigna news[3], 'Mekalh Tigray[4]' and 'wurayna'[5] journal. The contents of these sources are stored in a file and then verbs have been taken from these.

4.2 Data Annotation

As it was explained in the problem statement, Tigrigna is one of the under-resourced natural languages that have scarcity of data resources for research work. To the best of the searching and literature reviews done, a collected raw data and annotated dataset were not found. Therefore, the verbs were collected from different sources and then were annotated one by one manually. The annotation process includes the segmentation of the verbs and giving labels of the segmented parts(morphemes). The verbs are segmented based on prefix-stem-suffix manner where the labels are inserted at the end boundary of these parts (segmented morphemes). The labels have the meaning of the morphemes attached to the stem based on the subject and object marker that considers the number, gender and person. The passive, causative and negative prefix morphemes are also included in the segmentation. English capital letters are used for labeling which later are used as classes. For example, the verb 'ጨወየት' is segmented as 'ጨወየ[S] ት [A]'. The first part or 'ጨወየ' is considered as the stem labeled with S in the annotation as shown on Table 3.1below. The second part or 'ት' is the suffix that indicates a third person singular feminine subject labeled by the letter A in the annotation. A total of 26 different labels are used for all the training dataset. The annotated dataset is then stored in a file from which the instances are produced. The result of this process is then used as input to the preprocessing task. The following table shows examples of input verbs, their annotated input and the meaning of the label symbols.

[2] www.tigrigna.voanews.com
[3] https://www.bbc.com/tigrinya
[4] https://meqalh.com/
[5] http://www.wurayna.com/

Input verb	Labeled verb	Label meaning
ጨወየት	ጨወየ[S]ት[A]	ጨወየ[Stem]ት[third person singular feminine subject marker]
ተዘሪጉኩም	ተ[P]ዘሪጉ[S]ኩም[L]	ተ[passive prefix]ዘሪጉ[Stem]ኩም[second person plural masculine object marker]
ዘወረኒ	ዘወረ[S]ኒ[E]	ዘወረ[Stem]ኒ[first person singular object marker]
ዘወረኪ	ዘወረ[S]ኪ[H]	ዘወረ[Stem]ኪ[second person singular feminine object marker]
ዘሪጉና	ዘሪጉ[S]ና[G]	ዘሪጉ[Stem]ና[first person plural object marker]

Table 4. 1 : Examples of annotated data

4.3 Model Architecture

The following figure (Figure 4.1) shows the general architecture of the model that includes the training and the testing steps. In the training phase, the annotated verbs are taken as input to the preprocessing task in which the verbs are labeled to provide supervision during learning. The preprocessing step then prepares the input to learning step. This is to prepare the list of instances that are to be stored and will be used during classification processes when a new input is given to be classified. Each instance is a fixed length vector with features as elements of the vector. The next step in training phase is then to store the list of instances that uses memory based learning with instance-based learning algorithm. Then a learned model is produced which holds the instances and their feature weights. In the testing phase, surface verbs (normal inflected and unlabeled verbs) that are not used during training are taken as input to the preprocessing step. TiMBL[6] is used as training and testing tool which takes similar format of the instances. The preprocessing step then prepares the instances that will be used as input to the classifier. The classifier then calculates the similarity between the stored instances and a test instance to predict the class label of the input instance. The classification is at the instance level. The details of each step are described in the following sections.

[6] https://github.com/LanguageMachines/timbl

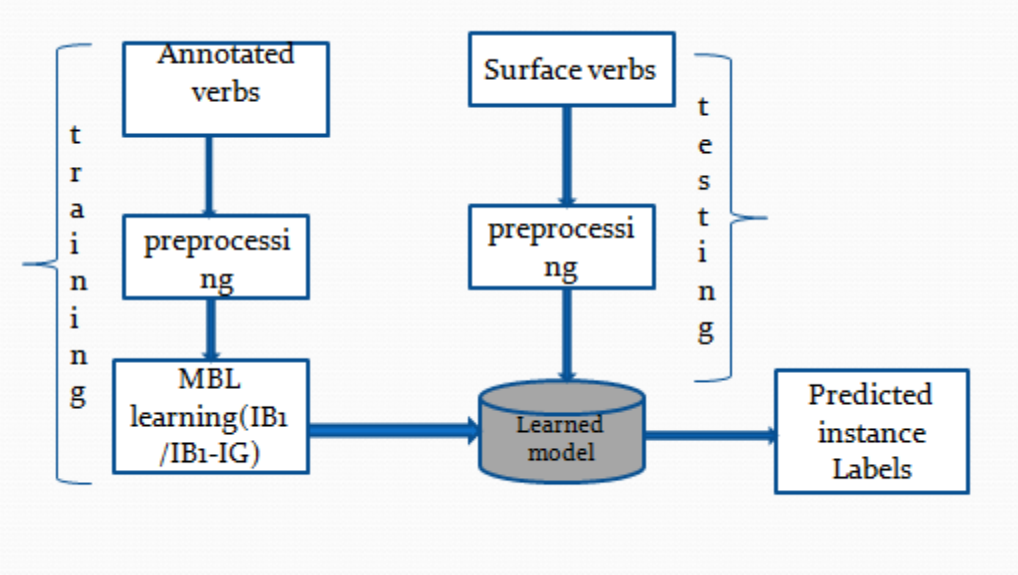

Figure 4.1: General architecture of proposed system

4.4 Preprocessing

The annotated data is not used directly to the learning step. It is preprocessed before the learning model accepts the input. The annotated data is first transliterated into Latin characters[1] as the learning tool does not support Tigrigna characters. The main preprocessing step prepares the list of instances that are used as training input lines taking the outputs of the annotation process. Each instance is a fixed-length feature-vector that has set of characters as elements (example, l,ä,t,ä,m,ä,n,a) of the vector filtered from the annotated data. In this work, TiMBL is used as learning tool. It is a memory-based learning tool that uses different data formats. In this case, C4.5 [28] data format, a comma separated character format, is used for each instance. The output of the preprocessing is, therefore, a C4.5 data format which is used as an input to the learning tool. Each input verbs produces as many instances as the number of letters used in the verb putting each letter in the middle of the vectors down by shifting the previous middle letter to left until the last letter become at the middle similar to[25]. The size of the feature vector is set to be thirteen excluding the class labels which are the maximum length out of the input data. The middle letter of the vector is the focus letter that has six left features and six right features (comma separated characters). The following algorithms show the general process of the preprocessing step that produces list of instances for the training. A filler input is used when there is no value for a given index. Appendix D Shows sample list of instances produced. In this work, a total of 26 classes are used in the whole dataset. These class symbols are listed at page *vii* (the list of symbols).

Input: *annotated verb*

Output: *list of Instances*

1. Define the vector to hold features.

2. read input from file into string

3. Fill starting feature at middle position and keep filling the characters until the right side vector is full.

4. If label symbol is found next to middle feature, put the symbol at class index

Otherwise put zero at class index

5. Shift the previous middle feature to left and start putting each feature as in step 3 making the

next feature at middle position

6. Repeat until all features become at middle position

7. update input

The following example illustrates the above algorithm:

Given the annotated verb, 'ለተመ[S]ና[G]' transliterated to 'lätämä[S]na[G]' as an input, the final preprocessed output is shown below where the '=' character is used when there is no feature value at that position(index).

*=,=,=,=,=,=,**l**,ä,t,ä,m,ä,n,0*

*=,=,=,=,=,l,**ä**,t,ä,m,ä,n,a,0*

*=,=,=,=,l,ä,**t**,ä,m,ä,n,a,=,0*

*=,=,=,l,ä,t,**ä**,m,ä,n,a,=,=,0*

*=,=,l,ä,t,ä,**m**,ä,n,a,=,=,=,0*

*=,l,ä,t,ä,m,**ä**,n,a,=,=,=,=,S*

*l,ä,t,ä,m,ä,**n**,a,=,=,=,=,=,0*

*ä,t,ä,m,ä,n,**a**,=,=,=,=,=,=,G*

The output of the above processes is then to be used as direct input for the learning method, particularly; the memory-based learning trained using TiMBL. As the main task of model is to predict the classes of the instances, the supervision is provided at the annotation and then prepared at the preprocessing making suitable for the learning tool. The preprocessing at the input of the classifier (testing) is similar except the input is not annotated and therefore the

classes are to be predicted by the classifier. The fourth step above is not included at preprocessing of the testing. The following steps are used to prepare the test instances.

Input: *verb*

Output: *list of Instances*

1. Define the vector to hold features.

2. read input from file into string

3. Fill starting feature at middle position and keep filling the characters until the right side vector is full.

4. Shift the previous middle feature to left and start putting each feature making the

next feature at middle position

5. Repeat until all features become at middle position

6. update input

4.5 Memory Based Learning

As it is explained in the background section, one of the learning mechanisms in machine learning is the supervised learning technique. In this work, memory-based learning is used which is one of the supervised learning techniques. Memory-Based Learning (MBL) is one of the techniques that has been proposed to learn different NLP classification problems. This learning technique, as explained in [28] is founded on the hypothesis that performance in cognitive tasks is based on reasoning on the basis of similarity of new situations to stored representations of earlier experiences, rather than on the application of mental rules abstracted from earlier experiences.

An MBL system, visualized schematically in Figure 4.2, contains two components: a learning component which is memory-based (from which MBL borrows its name), and a performance component which is similarity-based. The learning component of MBL is memory-based as it involves adding training instances to memory (the instance base or case base). An instance consists of a fixed-length vector of n feature-value pairs, and an information field containing the classification of that particular feature-value vector. The main task in the learning is therefore, storing the training instances and calculating the weighting of features used in training dataset.

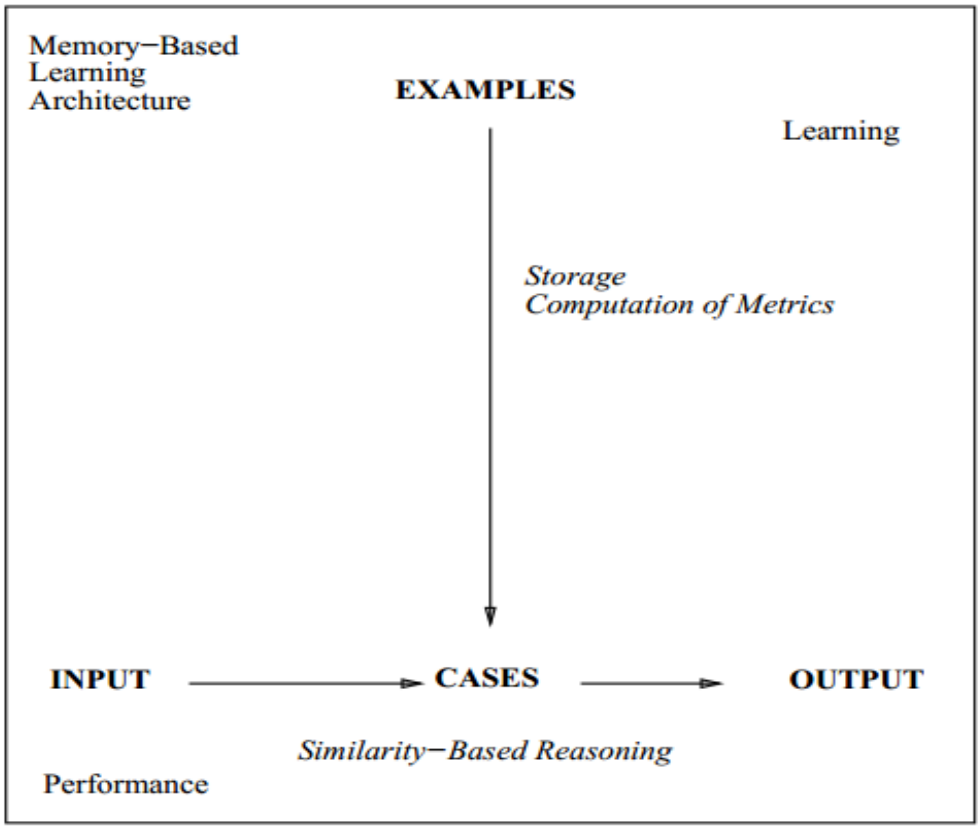

Figure 4.2: general architecture of memory-based learning system[28]

In the performance component of an MBL system, the product of the learning component is used as a basis for mapping input to output; this usually takes the form of performing classification. During classification, a previously unseen test example is presented to the system. The similarity between the new instance X and all examples Y in memory is computed using some distance metric $\Delta(X, Y)$ as explained in the section below.

4.6 Classification

The memory-based learning algorithm used in this work is the instance-based (IB1) learning algorithm. This algorithm is based on the nearest neighbor method that processes instances incrementally [29]. This algorithm uses similarity metrics that calculate the distance between the stored instances and a new instance given to it. The similarities are numeric-valued. The classification of a new instance then depends on the calculated similarity values where the most similar(smaller distance) are primarily considered as the class for the new instance that is decided based on a given decision method. In IB1, the feature relevance is not considered during classification where each feature of a given instance is to have the same contribution for the classification. The features in an instance are the set of characters used in the training data set. Another way is considered for the calculation of the similarity that considers the features during classification. This is all about weighting the features that will help to decide which feature

contributes better. The weighting used for this feature relevance is the information gain explained in the following section. This method is then known as the IB1-IG [25, 28]. The differences between the IB1 and IB1-IG is therefore only on the weighting of the features that are included in the calculation of the overlap distance metric. The IB1 does not consider the weighting while IB1-IG does.

Once the training inputs are stored as list of instances, the classification process uses similarity metrics to calculate how much similar the new input instance is with the stored instances. Having the distances or similarity calculated using the similarity metric; a decision should be taken by the classifier to assign the predicted class to the new input. New input verbs that are not seen during the training are received by the preprocessing task. The preprocessing creates a list of instances without having the class labels. The classifier then predicts the class labels of the new input instances using the distance metric used and the decision method for a class label.

4.6.1 Similarity Metrics

The similarity metrics are the methods used to match the feature values of the stored instances during the training and the new instances to be classified. It is all about computations of feature value matching [25].

Overlap distance metric: this metric is the most straightforward distance metric used in memory-based learning. In this metric, the identical feature values have an overlap distance of zero while the non-identical feature values have an overlap distance of one. This metric is considered as a best choice for symbolic feature representations. The distance between two patterns is simply the sum of the differences between the features. Equation (4.1) shows the calculation of the distance metric, where $\Delta(X, Y)$ is the distance between instances X and Y, represented by n features, and δ is the distance per feature. This algorithm is known as IB1 when there is no weighting for the features [28].

$$\Delta(X, Y) = \sum_{i=1}^{n} \delta(xi, yi) \qquad (4.1)$$

Where:

$$\delta(x_i, y_i) = \begin{cases} abs((x_i - y_i)/(max_i - min_i)) & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \qquad (4.2)$$

Weighted overlap metric: The overlap metric explained above simply counts the number of (mis)matching feature values in both patterns. If we do not have information about the importance of features, this is a reasonable choice. But if we do have some information about feature relevance one possibility would be to add linguistic bias to weight or select different features. An alternative more empiricist approach is to look at the behavior of features in the set of examples used for training. We can compute statistics about the relevance of features by looking at which features are good predictors of the class labels. Information Theory gives us a useful tool for measuring feature relevance in this way. Information gain is used as weighting to the features to assess their importance during classification. This algorithm is known as IB1-IG, instance-based with information gain [25].

Information Gain (IG) weighting looks at each features in isolation, and measures how much information it contributes to our knowledge of the correct class label. The Information Gain of feature f is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature [28]. The information gain is calculated using Equation 4.3.

$$\mathrm{Wi} = \mathrm{H(C)} - \sum_{v \in Vi} \mathrm{P(v)} * \mathrm{H(C|v)} \qquad (4.3)$$

$$\mathrm{H(C)} = -\sum_{c \in C} \mathrm{P(c)} \log_2 \mathrm{P(c)} \qquad (4.4)$$

where C is the set of class labels, H(C) is the entropy of the class labels , Vi is the set of values for feature i, and H(C| v) is the conditional entropy of the subset of the training examples that have value v on feature i. The probabilities are estimated from relative frequencies in the training set. The equation of the similarity metric, Equation 4.1 therefore becomes Equation 4.5 below. This algorithm is known as IB1-IG, instance based with information gain feature weighting.

$$\Delta(X, Y) = \sum_{i=1}^{n} \mathrm{Wi}\ \delta(\mathrm{xi}, \mathrm{yi}) \qquad (4.5)$$

Where Wi is the weight of the i^{th} feature

4.6.2 Decision

Once the instances are established in the model's knowledge base, how they are represented and how similarity between a test instances is computed, a final and equally crucial question concerns the nature of the decision of the labels to the new instances, that is, how a class is assigned to novel instances given its similarity to each instance in memory. In other words, the decision determines how the class voting process is done using weighting the distances from the

target classes to be classified. Two mechanisms are used for the class voting. The majority class voting and inverse distance weighted method. The neighbors are selected based on the distance. Therefore, the parameter k is interpreted as number of nearest distances [27].

Majority Class Voting:

The most straightforward method for letting the k-nearest neighbors vote on the class of a new case is the majority voting method, in which the vote of each neighbor receives equal weight, and the class with the highest number of votes is chosen. It is the basic method used to determine the class to be assigned to a query. It consists of doing a frequency count of the classes occurring in the nearest neighbor set, and then assigning the most frequently occurring class to the query. This means that all instances in the set of nearest neighbors contribute equally to the decision about which class to assign to the query, irrespective of their distances from the query point [27].

Inverse distance weighting: the second method is the inverse distance, that is, the neighbors of a query are assigned the weight reciprocally to the distance to vote for the predicted class. The inverse distance method can be formulated as in Equation 4.6 with w_j , the weight of j^{th} neighbor, d_j , the distance of j^{th} neighbor from the unknown class and e , smaller number that avoids division by zero:

$$w_j = 1 /(d_j + e) \qquad (4.6)$$

This distance weighting function assigns highly differing weights for close neighbors, and less differing weights for more distant neighbors. ID assigns very high votes (distance weights) to nearest neighbors at distances approaching 0[28].

CHAPTER 5

Result and Discussion

5.1 System Description

Morphological analysis of a natural language consists of many tasks that can be described separately and applied for different end task applications. In this system, the main task of the analysis is the segmentation of a given Tigrigna verb into its smaller meaningful parts or morphemes. The system is a machine learned model that is trained using a supervised type of learning particularly using a memory based learning algorithm. The knowledge base for its generalization is the stored instances that are used for predicting the labels of a new input verb to detect the correct segmentation boundary. It uses English capital letter characters to detect the boundaries of the morphemes that it predicts the labels and transfers to the new input instances. The instances are produced through the preprocessing task which deconstructs the input verbs into comma separated characters. Each instance is then used as an input line to the classifier where the distance from each stored instances is calculated using the distance metrics and the decision methods for the prediction described in chapter four. The system mainly checks whether the segmentation labels occur at the correct position of the input verb given to it or not. It was trained using 4416 input instances and tested using 1328 instances extracted from 718 annotated Tigrigna verbs taken from the collected corpus of verbs.

To illustrate how the system does the classification task, the instance, =,=,z,ä,y,r,u,k,i,=,=,=,=,? , with unknown class can be taken. The question mark here is to indicate the unknown class. The instance is first prepared using the preprocessing task. This instance is given to the classifier as an input and then the classifier calculates the distance between all the instances stored during training which are similar to the instances at appendix D using the overlap metric discussed in section 4.6.1. Then it selects the nearest instances according to the calculated distance by setting the k values. The decision to assign the final class is then done based on the decision methods described in section 4.6.2.The following table illustrates the different results of the output taken from the output file. The incorrect result below indicates that the labels predicted are not the correct class labels whereas the correct one indicates the labels correctly segmented the morphemes and are correctly predicted.

Input verb	Correctly analyzed	Input verb	Incorrectly analyzed	Corrected to
zäwärätə,ዘወረት	[zäwärä]S[tə]A, [ዘወረ]S[ት]A	lätämäna,ለተመና	[lätämä]S[na]W, [ለተመ]S[ና]W	[lätämä]S[na]G
ayənäbäränə, ኣይነበረን	[ayə]Z[näbärä]S[nə]B, [ኣይ]Z[ነበረ]S[ን]B	läkämäna, ለከመና	[läkämä]S[na]W, [ለከመ]S[ና]W	läkämä]S[na]G

Table 5. 1: Sample output

Evaluation method: The performance of the model was evaluated using accuracy which is the percentage of correctly predicted instances of the total test input instances. It is calculated using the following equation.

$$\text{Accuracy}(\%) = \frac{correctly\ \ predicted\ \ instances}{\text{total test input instances}} * 100 \qquad (5.1)$$

This equation can be illustrated with an example: the total test input instances are 1328 and if the 1020 of these instances are correctly predicted by the model, then the accuracy in percent would be equal to (1020/1328)*100

5.2 Experimental Setup Tools

The model was implemented using Intel(R) core i3 processor of speed 2.40 GHz computer with 4GB RAM and 500GB hard disk on Linux Operating system(Ubuntu distribution) of version 12.04 LTS. TiMBL[28] was used as learning and testing tool that has robust functionality for memory based learning tasks. It is implemented with C++ programming language that is used for discrete value classifications mainly for natural language processing tasks. Appendix E shows a sample output of testing process from TiMBL. The following table summarizes the tools used in this work.

Hardware tools:

Hardware	**Description**
Hard disk	500GB
RAM	4GB
Processor	Intel Corei3

Table 5. 2: Hardware tools used in the experiment

Software tools:

Software and related tools	**Description**
OS	Linux (Ubuntu distribution) version 12.04 LTS
TiMBL	Learning tool
Netbeans IDE version 8.1	Programming environment
Java	Programming used for implementing preprocessing algorithm

Table 5. 3: Software tools used in experiment

5.3 Result

The experiments were done based on the overlap metric (the overlap metric without weighting the features and with information gain weighting features) and decision methods described in Chapter Four. The number of nearest neighbors, k, and parameter were used by varying from the smallest (one) up to 17. A total of 68 experiments were done as shown on the tables below. The execution time taken during the classification was also recorded for the different parameters of the nearest neighbors.

Result obtained using overlap metric (IB1) and majority class voting decision method:

The following table shows the result obtained using the overlap metric without weighting the features for 17 variations of the number of nearest neighbor parameters, k.

Number of nearest neighbor(k)	Accuracy in %	Time taken in seconds
1	91.56	0.1313
2	87.65	0.1802
3	85.84	0.2308
4	86.14	0.2803
5	84.04	0.3288
6	81.63	0.3718
7	78.91	0.4200
8	79.22	0.4459
9	78.61	0.4744
10	78.31	0.4963
11	78.31	0.5047
12	78.31	0.5144
13	78.31	0.5228
14	78.31	0.5298
15	78.31	0.5219
16	78.31	0.5202
17	78.31	0.5233

Table5. 4 : Accuracy and execution time with respect to k parameters for IB1 with majority voting

As it can be seen from the result on Table 5.4 above, the highest accuracy was obtained when the number of nearest neighbors was set to be one. The accuracy decreased with increasing the number of nearest neighbor or k values in general. This indicates that when the number of neighbor increases, the misclassification of the class labels increased in the case of the features used in this work. However, the accuracy was saturated starting from the k value of 9 up to 17 which showed total saturation at k values starting from 10. The increment of k values was stopped as it showed saturation of the accuracy. The accuracy at k=3 and k=4 showed a reverse result that the accuracy is higher at k=4 than at k=3 which increased at the higher k value though

the change is smaller. But in general, the accuracy decreased with increasing the k values. The saturation indicates that the number of misclassification becomes smaller and it can also be stated that the highest differences that cause the misclassification occurred at those k values.

The execution time also showed variations with different k (number of nearest neighbors) values as shown from Table 5.4 above which showed a total increment with the increment of the k value. The time complexity is directly related to the number of test input as explained in [28]. However, when it was observed even with the same input, the execution time increased with the increment of k values. This indicates that the number of nearest neighbors also affects the execution time during classification. However, the execution time increased even when the result was saturated at the highest k values as shown from Figure 5.2. This can indicate that there is an overhead on the execution time irrespective of the saturation of the accuracy. The execution time was recorded and discussed due to the observation during the experiments though execution time is more related with the algorithm(classifier) used as it was explained in different research results[28]. However, it was found that the execution time also matters when an algorithm is used. But time for classification is one constraint of memory-based algorithms in general [28]. A graph was drawn for the variations of the number of nearest neighbors with respect to execution time and accuracy for general observation of the variation. Figure 5.1 and 5.2 show these variations.

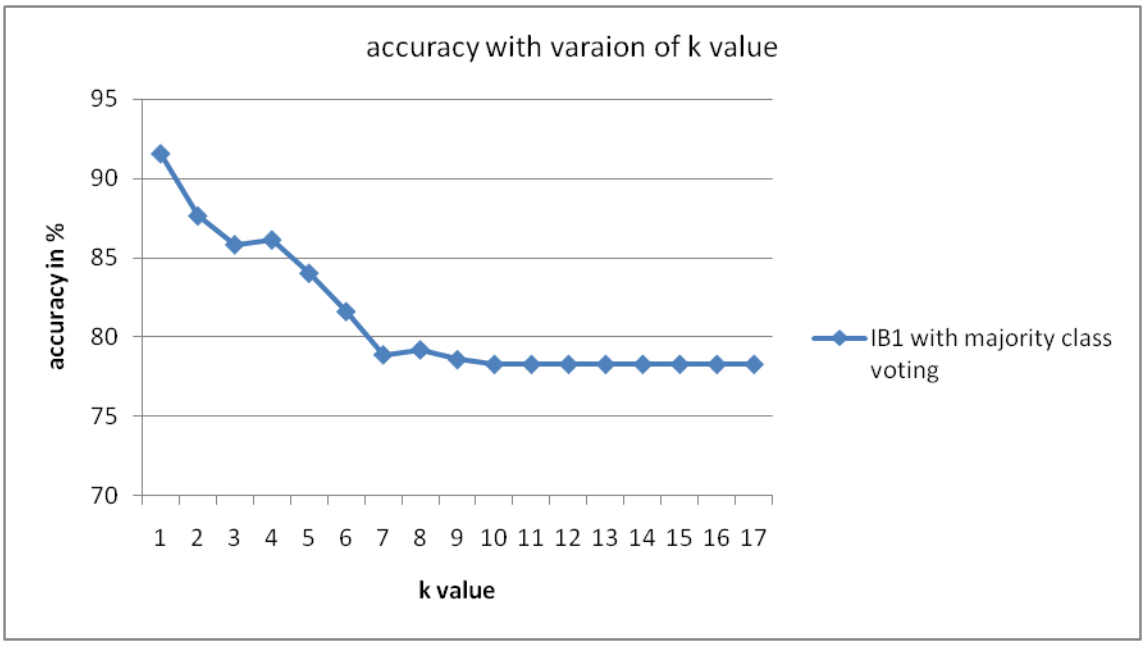

Figure 5.1: Accuracy of IB1 using majority class voting with variation of k value

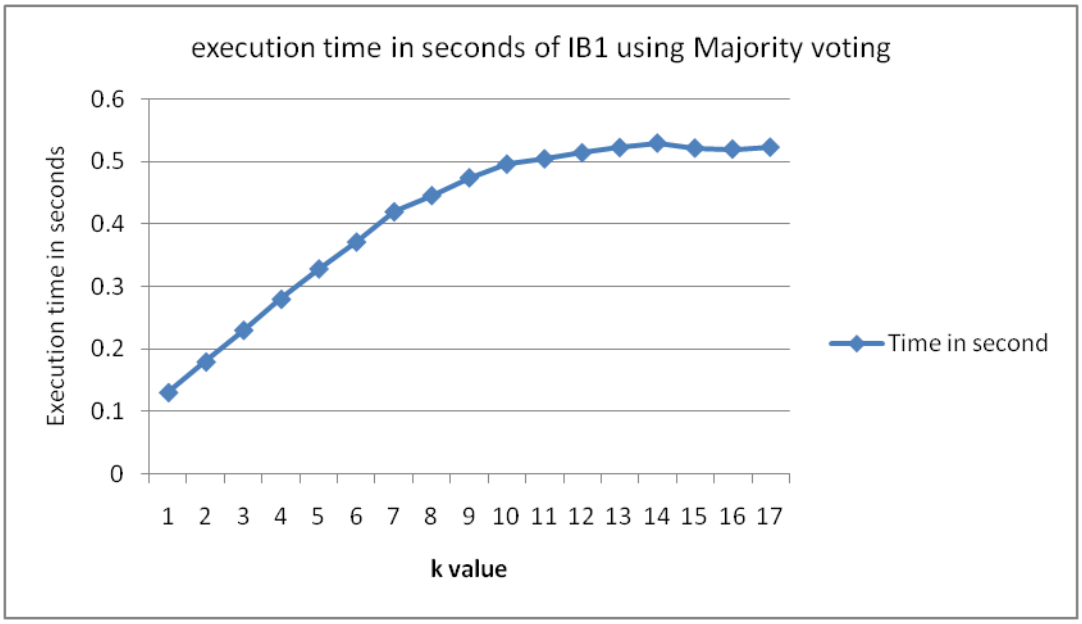

Figure 5.2 Execution time of IB1 using majority class voting in seconds with variation of k value

Result of IB1-IG using majority voting decision method: Table 5.1 shows the result obtained using the overlap metric without considering the feature relevance. In other words, the features used in the training have the same relevance for classification. Whereas table 5.2 below is the result obtained considering the feature relevance based on information gain as described in Chapter Four sections 4.5.1.

Number of nearest neighbors(k)	Accuracy (%)	Time taken in seconds
1	88.55	0.0960
2	89.15	0.1225
3	87.05	0.1442
4	86.44	0.1507
5	86.44	0.1766
6	86.44	0.1856
7	85.24	0.1906
8	85.24	0.2015
9	84.33	0.2141
10	83.43	0.2286
11	84.33	0.2398
12	84.03	0.2530
13	84.94	0.2700
14	84.33	0.2883
15	83.43	0.2836
16	82.53	0.2932
17	82.23	0.3021

Table 5. 5: Accuracy and time with respect to k parameters for IB1-IG with majority class voting

For the weighted overlap metric, the result shown above was obtained for seventeen values of the nearest neighbor parameter, k. The decision method for final assignment of the classes used here is the majority class voting where the most occurring class is taken as class of the unknown input instance. As it can be seen from table 5.2 above, the highest accuracy was achieved when the number of nearest neighbor parameter was set to be two. The result shows that the accuracy has a decrement when the number of nearest neighbor was increased except for some fluctuations that reversed the accuracy value and some are constant while increasing the parameters as shown from the table above. For example for value of k from 11 up to 14, the accuracy was almost constant which indicates that the rate of misclassification for those parameters is almost similar. In general, the number of nearest neighbors affected the result as it can be seen from the table where the minimum accuracy was achieved with the maximum k value. The result has higher fluctuation when it is seen from the minimum to the maximum accuracy.

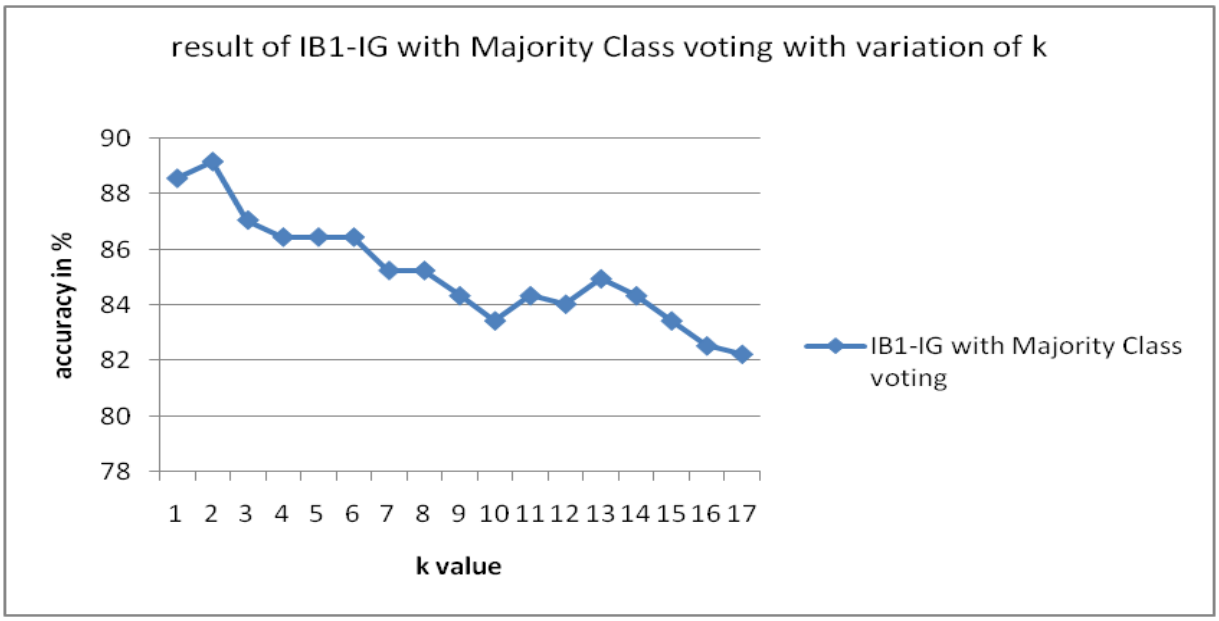

Figure 5.3: accuracy of IB1-IG using majority voting

The execution time was also recorded for each k value similar to the result in table 5.1. As it can be seen from Table 5.2, execution time increased with increasing the nearest neighbor parameter (k) value. This indicates that the number of nearest neighbors affects the execution time in addition to the effect for the accuracy. Fortunately, the better accuracy was obtained with the smaller execution times as shown from the table above. The overhead would obviously be worst when the data set is huge.

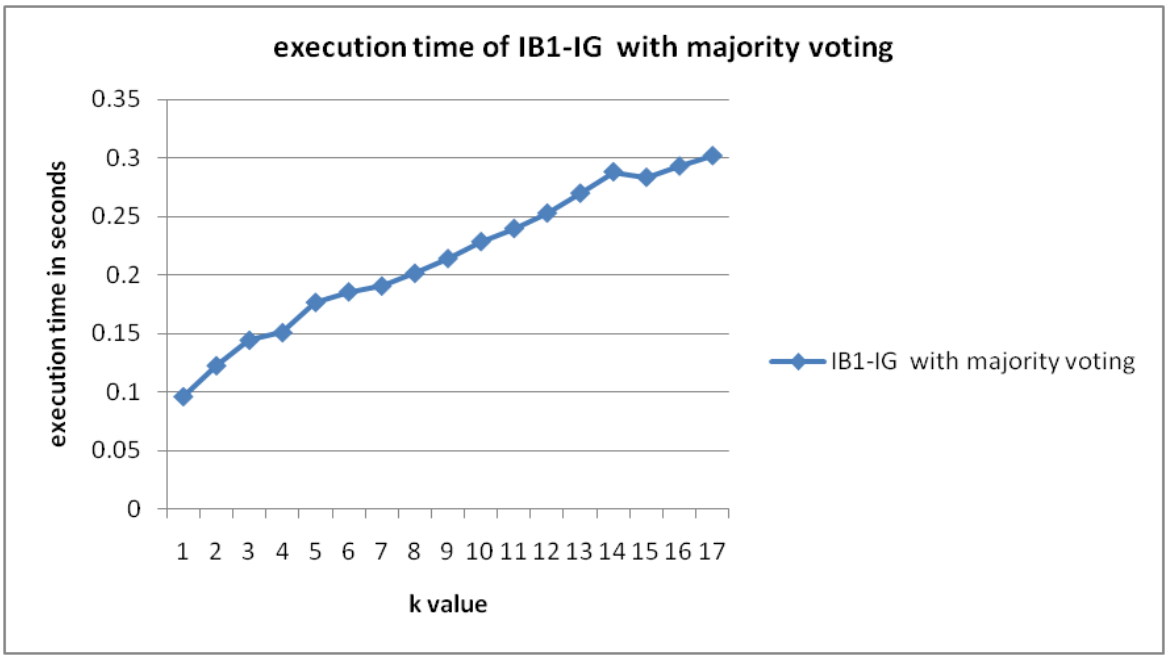

Figure 5.4: execution time of IB1-IG with majority voting

Results of IB1 with inverse distance weighting:

The results recorded in Table 5.1 and Table 5.2 is based on the overlap metric without weighting the features and with weighting the features using information gain. These experiments were done using the same decision method of the final assignment for the classes of the new input. The following result also is based on inverse distance weighting as described in chapter four. This method assigns highest weight for the nearest one calculated using Equation 4.5. Tables 5.6 and 5.7 show the results of this method for both the overlap without weighting the features and with weighted overlap (IB1 and IB1-IG) respectively.

Number of nearest neighbors(k)	Accuracy (%)	Time taken in seconds
1	91.56	0.1313
2	89.15	0.1793
3	88.25	0.2266
4	88.55	0.2762
5	87.65	0.3226
6	85.84	0.3694
7	81.92	0.4180
8	81.92	0.4549
9	81.32	0.4761
10	81.32	0.4967
11	81.32	0.5088
12	81.32	0.5230
13	81.32	0.5276
14	81.32	0.5241
15	81.32	0.5285
16	81.32	0.5209
17	81.32	0.5220

Table 5. 6: Accuracy and time with respect to k parameters for IB1 with inverse distance weighting

As it can be seen from Table 5.6 above, the experiment for IB1 with inverse distance weighting were simulated for 17 k values where the highest accuracy was obtained at the smaller k value,1. The accuracy changed with smaller amount for the k value starting from 2 up to 6 as shown from the table. However, the accuracy suddenly lowered from k value of 7 where it started to be in a saturation point. It saturated for about ten variations of the nearest neighbor starting from the seventh value. This indicates that the rate of misclassification at those k values is almost constant. But in general, the accuracy showed a smooth decrement with the increment of the nearest neighbor parameter where for the last 11 k values become constant. The differences between the minimum and maximum values is somehow large(10.42) that indicates the number of nearest neighbors highly affects to the accuracy. Figure 5.5 and Figure 5.6 show the whole trends of the accuracy and execution time respectively.

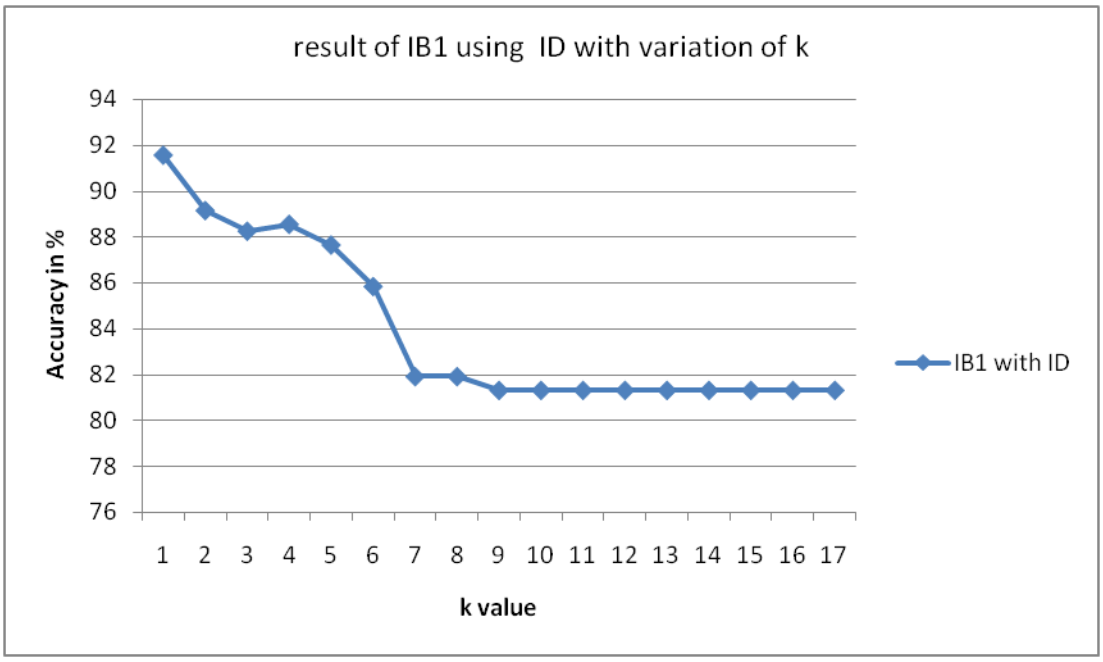

Figure 5.5: accuracy of IB1 using ID with variation of k

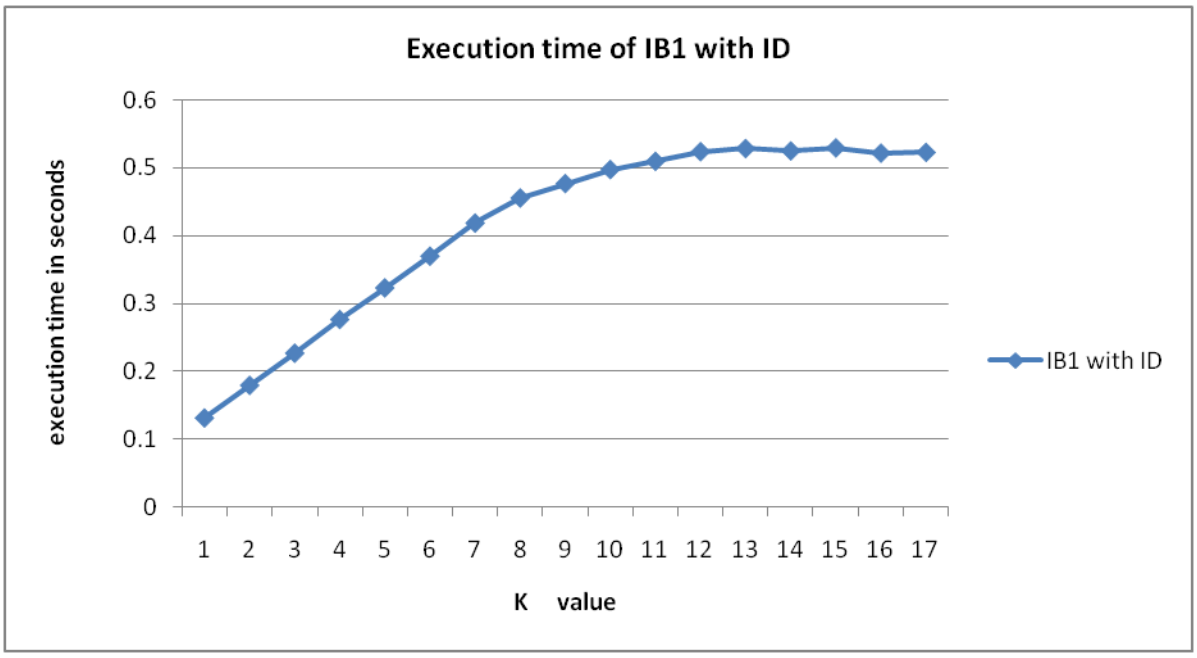

Figure 5.6: execution time in seconds of IB1 with ID

The execution time also has a general increment with the increment of the number of nearest neighbors or k value as shown from the table above. Similar to the results shown in Table 5.1, the best result (accuracy) was obtained at the minimum execution time. The time increment is higher at the higher increment of the accuracy but did not reach a saturation point like the

accuracy. This indicates that when the accuracy has smaller changes at the increment of the k values, the execution time also have smaller changes. In general the selection of number of nearest neighbor has an effect on both the accuracy and the execution time.

Result of IB1-IG using Inverse Distance:

Number of nearest neighbors(k)	Accuracy (%)	Time taken in seconds
1	88.55	0.0959
2	87.65	0.1182
3	88.25	0.1506
4	88.25	0.1504
5	88.25	0.1617
6	88.25	0.1814
7	87.65	0.1855
8	87.35	0.2103
9	87.35	0.2120
10	87.05	0.2257
11	87.65	0.2339
12	87.95	0.2435
13	87.65	0.2584
14	87.65	0.2680
15	86.75	0.2756
16	86.44	0.2864
17	86.74	0.2990

Table 5. 7: Accuracy and time with respect to k parameters for IB1-IG with inverse distance weighting

Table 5.7 shown above is the result obtained using the weighted overlap distance metric and an inverse distance weighted decision method. As it can be seen from the table, the best accuracy was achieved at the smaller k value. The accuracy fluctuated between 86 to 88 with smaller decimal differences.The increment of the number of nearest neighbor did not bring large differences on the accuracy but little higher at smaller k values in general. This indicates that the instance based with weighted overlap metric with information gain has smaller fluctuation on the accuracy when the number of nearest neighbors varies.

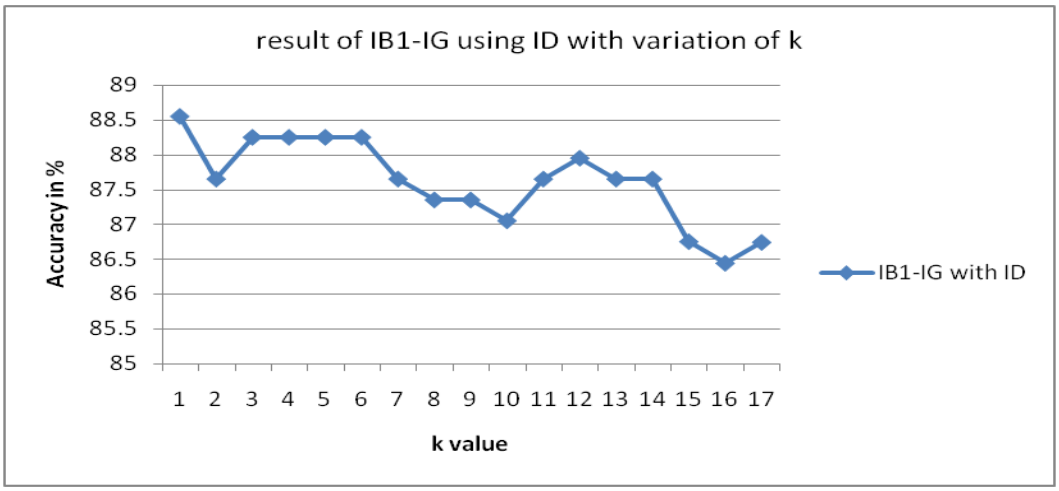

Figure 5.7: Accuracy of IB1-IG using ID with variations of k

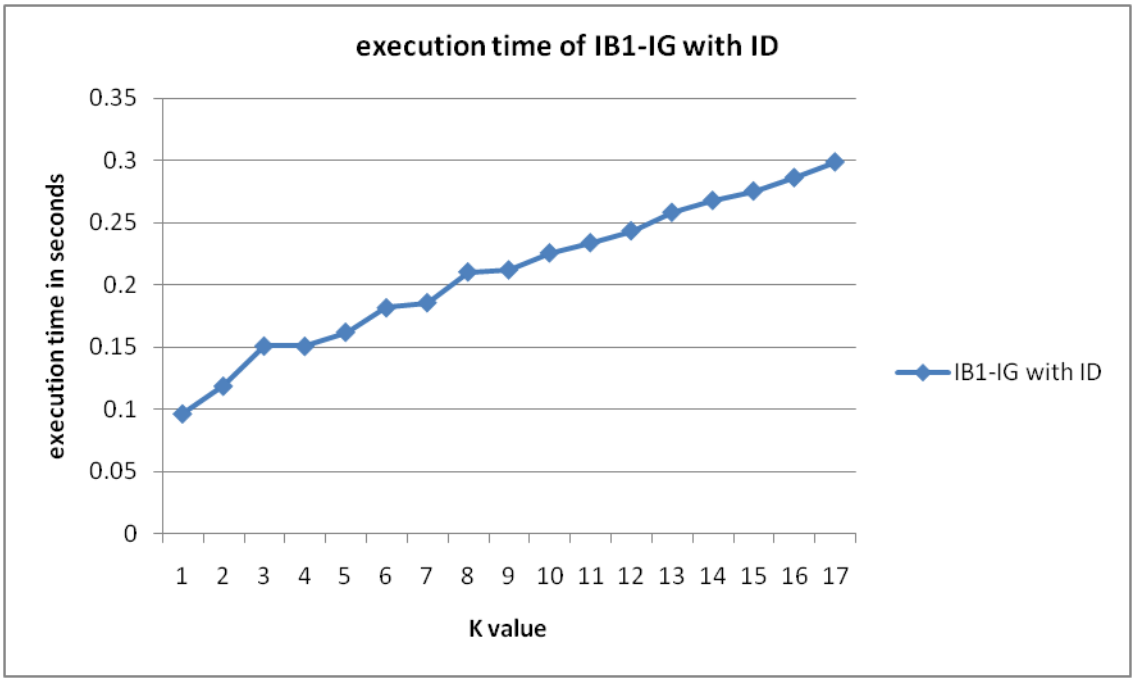

Figure 5.8: Execution time of IB1-IG with ID

The execution time also has an increment with the increment of the number of nearest neighbors as it can be shown from the table above (Table 5.7). The best accuracy was achieved with the smaller execution time. In general, the k value affects the execution time that brings higher overhead with the higher k values.

5.4 Comparing the Results

It is important to compare the results that are obtained from the metrics (the weighted and non-weighted overlap) and decision methods simulated with the variation of the number of nearest neighbors to identify the best metric and parameter (k value). The result of the two decision methods (the majority class voting and the inverse distance weighted class voting) with weighted and non-weighted overlap metric are compared for the accuracy and execution time for their best and overall results.

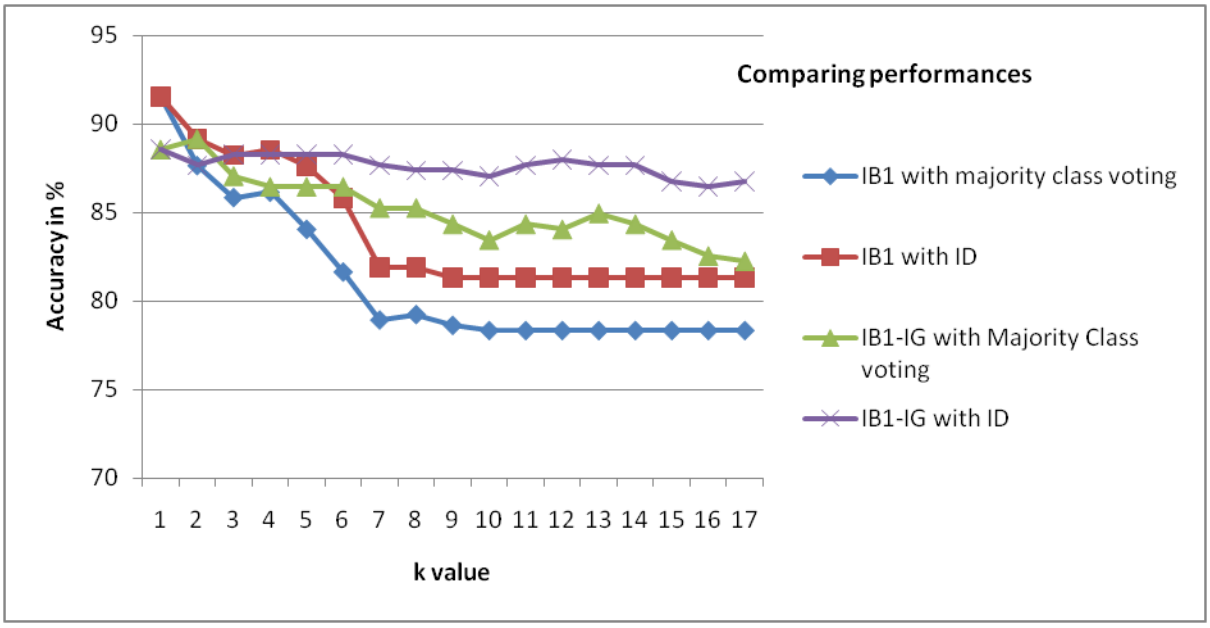

Figure 5.9: comparison of accuracies of all experiments

As it can be seen from the graph in Figure 5.9, the highest accuracy was achieved with overlap metric using inverse distance weighting (IB1 with ID) and the overlap metric using the majority class voting at the smallest k value. However, the performance decreased as the k value increased in the case of majority class voting decision method compared to the inverse distance weighted decision method. This indicates that irrespective of the distance metric, both decision methods have the same performance when the smallest number of nearest neighbor parameter is used but when the parameter becomes higher and higher, the misclassification increased in the majority voting method as the farthest neighbors are considered same with the very near ones which causes to decrement of the accuracy. The inverse distance class voting decision method is

therefore better than the majority voting for all the k values used when the performance for both methods are generalized for segmentation of Tigrigna verbs as indicated by the legends.

When we look at the weighted overlap metric(weighted with information gain) using the two decision methods, the majority voting method achieved better at the smallest k values particularly at k value of 2 but equal when the k value is the smallest(1) as shown from the tables and the graph above. However, the accuracy decreased (by 4.21) for the case of majority voting for the highest k values especially after the third value. This indicates that the feature relevance is important at the highest k values when inverse distance is used compared to the majority voting decision method.

When the weighted and non-weighted overlap methods using the same decision method are compared, the non-weighted overlap metric achieved better at lower k values particularly at the smallest k value but decreased worse than the weighted one at the higher k values as shown from the tables clearly when majority voting was used. The non-weighted overlap metric resulted in highest fluctuation when they are compared from minimum to maximum result within the used k values. The misclassification decreases when higher k values are used and the feature relevance is considered with their information gain. The non-weighted overlap achieved better at the first two k values but highly decreased at the highest k values and showed high fluctuation compared to the weighted overlap using the inverse distance class voting decision method. The latter showed small fluctuations from the minimum to the maximum k value that indicates still the relevance of the features matter when highest k values are considered.

When the execution times with the variations of the number of nearest neighbors are compared, two groups (IB1 with inverse distance and majority voting have similar results and IB1-IG with these two decision methods also have similar results) were found similar with smaller differences as shown on the following figure (Figure 5.10). But all the execution time of the IB1 and IB1-IG with the two decision methods increased with the increment of the k (number of nearest neighbors) value. The increment became smaller at the highest k values where the minimum accuracies were achieved. The execution time for the IB1-IG was generally better (smaller) compared to IB1(a differences of 0.0353 at minimum and 0.2212 at maximum k values) with the two decision methods which can indicate that the relevance of the features considered with the information gain helps search the nearest neighbors fast.

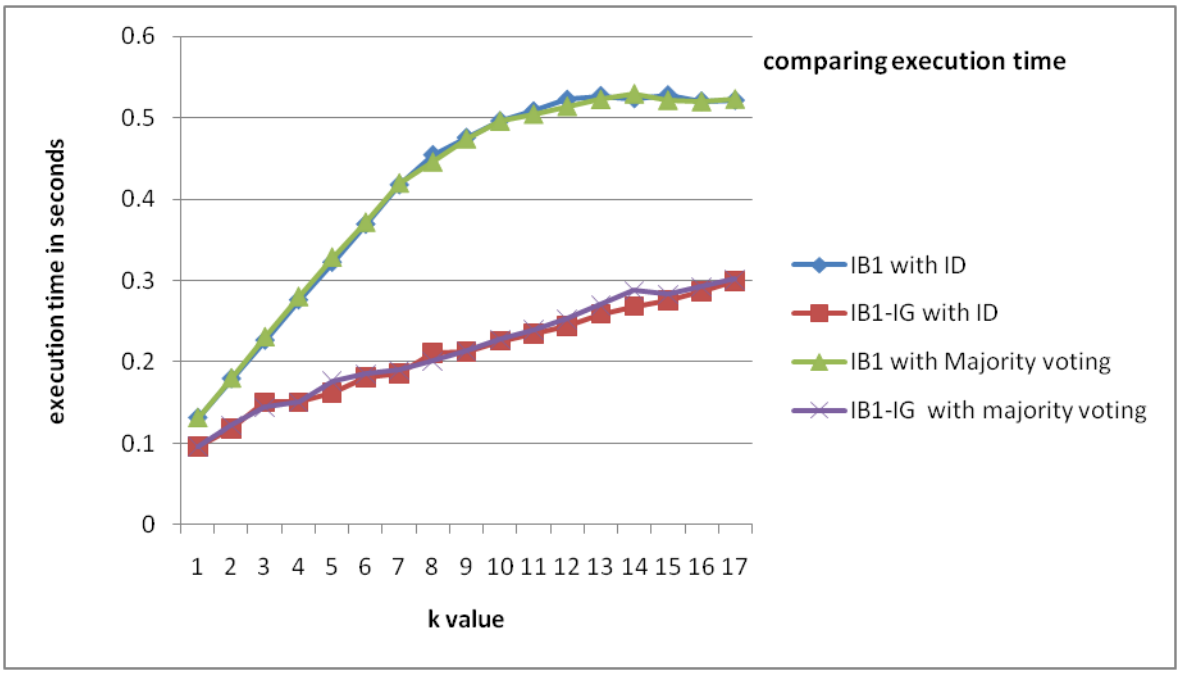

Figure 5.10: execution time of all experiments

Answer to Research question:

Question : can the instance-based algorithm be applied for morphological analysis of Tigrigna verbs?

Answer: Yes, the instance-based algorithm can be applied for the morphological analysis of Tigrigna language verbs as encouraging result was achieved that is described on the tables in the previous pages. A highest accuracy of 91.56% can indicate that the instance-based algorithm applies well for the detection of the morpheme boundaries of Tigrigna inflected verbs. Most of the other part of speeches in Tigrigna has similarity as most of them are derived from Tigrigna verbs. This can indicate that the method or algorithm used here can also be applied for the other parts of speech.

5.5 Summary

The experiment was done with different decision(class voting) methods and with the overlap distance metric with weighting and without weighting the feature relevance. The best results were obtained at the minimum number of nearest neighbor's parameter in general. The experiments showed that the feature relevance has much effect on the accuracy and execution time. The overlap metric without weighting achieved better results when smaller k values were used but had higher variation as k value increased. For this case the distance weighted decision

methods had also contribution where the inverse distance showed smaller fluctuations of the accuracy with increasing the k values compared to majority voting decision method.

The following table (Table 5.8) shows the best results obtained from the different experiments based on the metric and decision methods used. The numbers in the table (Table 5.8) are the accuracy obtained with the corresponding methods.

Method(algorithm)	decision methods	
	Majority voting	Inverse distance
IB1	91.56%	91.56%
IB1-IG	89.15%	88.55%

Table 5. 8: optimum accuracy of IB1 and IB1-IG

.

CHAPTER 6

Conclusion and Recommendation

This chapter summarizes the main points of the work including the methods used, the experimental processes, the result obtained from the experiment and recommendations for further work. Generally, it concludes the final work results and further possible works on the area.

6.1 Conclusion

The main goal of this work was to automate the morphological analysis of Tigrigna language verbs using machine learning technique which segments the morphemes of an input verb by detecting the boundaries of the morphemes. Memory-based learning technique, particularly, the instance-based algorithm was used to build the model where stored examples are used as knowledge base for its generalizations. To the best of our knowledge, data for the model training and testing were not available and hence the verbs were collected from scratch manually from different data sources. We annotated the collected data manually and prepared as instances collected data set were annotated manually and these annotated data was prepared as instances to be used as input to the training algorithm using a preprocessing algorithm implemented with java programming language. A total of 4,416 instances for training and 1,328(30 percent of the total) instances for testing were used. Though the data set is limited to this amount as the manual process was time consuming for both the collection and annotation tasks, it will be used as additional resources for a research work in similar areas which helps to minimize the problem of data scarcity of the language. To train and test the model, Tilburg Memory-Based Learner tool was used on Linux operating system.

The encouraging results obtained showed that the instance based algorithm achieved good accuracy in the predictions of the class labels that detects the boundaries of the morphemes of a given Tigrigna verb for its segmentation. An optimum accuracy of 91.56% was achieved with the overlap distance metric without weighting and with both decision methods (the majority voting and inverse distance methods). The simulations showed that the number of nearest neighbor's parameter affected both for the accuracy and execution time during classification. The weighting of the features showed a small difference of accuracy fluctuations compared to the non-weighted metric for the same decision method with the variation of k value. Though the execution time is one of the constraints of memory based algorithms, the variation of the number of nearest neighbors also affected it which showed that the higher value of number of nearest neighbor has greater effect. Tigrigna verbs have complex and different appearing formats which makes it difficult for its complete analysis. The alteration of letters or spelling changes and deletion or addition of letters of the verbs during analysis is not handled as it is very huge work to annotate them which needs more time. This causes for the limitation of coverage of the verbs.

In fact the annotation process was so challenging that caused for the limitation in the analysis. The future work of this thesis is the analysis of the whole morphology of the language that will have large coverage of the morphology of Tigrigna language.

6.2 Recommendation

The following main points are recommended as further works that can enhance and extend the current work and other similar works.

- A fully featured and well organized huge data set of the morphology of the language is needed which helps to work in preparing the training and test data set.
- The current work can be extended to include the analysis of verbs inflected or derived due to the infix.
- It can be extended to the other morphology of the language such as nouns and adjectives.
- Improving the performance of system can also be recommended by investigating other methods.

References

[1] D. Khurana et al., "Natural Language Processing: State of the art, current trend and challenges,"[online]Available:https://arxiv.org/ftp/arxiv/papers/1708/1708.05148.pdf. [Accessed Dec 10,2017]

[2] S. R. Anderson," Encyclopedia of Cognitive Science," [online]. Available: https://cowgill.ling.yale.edu/sra/morphology_ecs.htm [Accessed Dec 23, 2017]

[3] S. Norrby," *Using Morphological Analysis in Information Retrieval System for Résumés*,", , Msc. Thesis,Dept. CSc. and Commn., Sweden KTH,2016

[4] D. Jurafsky , J. H. Martin. , Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2nd ed., New Jersey :Pearson prentice Hall, 2008

[5] " Machine learning vs natural language processing," [Online]. Available: https://www.lexalytics.com/lexablog/2012/machine-learning-vs-natural-language-processing-part-1. [Accessed Nov 12,2017]

[6] Wikipedia, "Machine learning," [Online]. Available:https://en.wikipedia.org/wiki/Machine_learning [accessed Nov 20, 2017]

[7] B. Alemu "A Named Entity Recognition for Amharic," MSc. Thesis,Dept. Info.Science,A.A Univ. A.A, Ethiopia , June, 2013

[8] M. Selvam , A.M. Natarajan, "Improvement of Rule Based Morphological Analysis and POS Tagging in Tamil Language via Projection and Induction Techniques," *International Journal of Computers* ,Volume 3,Issue 4 , 2009

[9] M. Yonis, "Development of Morphological Analyzer for Af-Somali,"MSc. Thesis,Dept. Comp.Sc, A.A Univ., A.A, Ethiopia, Jun 2017

[10] T. Bati, "Automatic Morphological Analyzer for Amharic, An Experiment employing Unsupervised Learning and Auto segmental Analysis Approaches, "MSc. Thesis,Dept. Info. Sc, A.A Univ., A.A, Ethiopia, Jun 2002

[11] W. Mulugeta , M. Gasser*," Learning Morphological Rules for Amharic Verbs Using*

Inductive Logic Programming," Workshop on Language Technology for Normalization of Less-Resourced Languages, pp. 7-12,2012

[12] Y. Lee ," *Morphological Analysis for Statistical Machine Translation*," proceedings of HLT-NAACL 2004:short papers , Boston, Massachusetts, May 02-07, pp. 57-60, 2004

[13] T. S. Bellomo ,"*Morphological Analysis and Vocabulary Development: Critical Criteria*," The Reading Matrix , Volume 9, Number 1, April 2009

[14] J. Allen, Natural Language Understanding, 2nd ed. California:The Benjamin/Cummings., 1995

[15] J. Goldsmith, "*Unsupervised Learning of the Morphology of a Natural Language*," Computational Linguistics 27(2): 153 – 198,2001

[16] Kazakov et al., "*Unsupervised Learning for Word Segmentation Rules with Genetic Algorithms and Inductive Logic Programming*," Machine Learning, 43, 121–162, Kluwer Academic Publishers, 2001

[17] K. Koskenniemi ,"Two-Level Morphology: A general Computational Model for Word-form recognition and production," *proc. Of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics* , Stanford, California , July 02 - 06, 1984, PP. 178-181

[18] Y. Fissaha, "Development of Stemming Algorithm for Tigrigna Text," MSc. Thesis, Dept. Info. Sc. ,A.A Univ., A.A, Ethiopia, June 2011

[19] R. P. Bhavsar, B. V. Pawar, "*Rule based Word Morphology Generation Framework*," International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011

[20] M. Gasser, "*HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya*," Conference on Human Language Technology for Development, Alexandria, Egypt, 2-5 May 2011, pp 94-99

[21] J. Gold smith,"*An Algorithm for Unsupervised Learning of Morphology*," Natural Language Engineering 1 (1): 000–000. , Cambridge University Press, United Kingdom, Oct 2005

[22] X. TANG,(2006,Oct). "*English Morphological Analysis with Machine- learned Rules*," Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation,Y06-1005,2006

[23] A. Spencer ,"*Morphological Theory: An introduction to word structure in generative grammar*," Blackwell Publishers, Volume 28, Issue 2, pp. 509-512 ,Sep 1992

[24] D. Teklu, ዘበናዊ ሰዋስው ቋንቋ ትግርኛ (modern Tigrigna Language Grammar), 5th ed. , Addis Ababa : mega printing p.l.c , 2017

[25] A. v. den Bosch , W. Daelemans, "*Memory-Based Morphological Analysis*," Proc. the 37th Annual Meeting of the ACL, University of Maryland, pp. 285-292, Jun 20-26, 1999

[26] J. Zavrel , W. Daelemans , "*Memory-Based Learning: Using Similarity for Smoothing* ," in: Proc. 35th Annual Meeting of the Association of Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the ACL, pp.436-443, Madrid, Jul 1997

[27] E. Keuleers, "Memory-based learning of inflectional morphology," . Ph.D. thesis, University of Antwerp, 2008

[28] W. Daelemans et al., "TiMBL: Tilburg Memory-Based Learner version 6.4 ,Reference Guide, ILK Technical Report ,",Tilburg Univ.,Tilburg, *ILK 11-01,Jan 2018.*

[29] A. Kilber,"*Instance-based Learning Algorithms*,"Machine Learning, Volume 6, Issue 1, pp 37–66, Jan 1991

Appendices

Appendix A: writing Scripts(letters) of Tigrigna language

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሏ				
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ሗ				
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ	ሟ				
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ሧ				
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ	ሯ				
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ሷ				
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ	ሿ				
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
ቐ	ቑ	ቒ	ቓ	ቔ	ቕ	ቖ	ቘ	ቚ	ቛ	ቜ	ቝ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ				
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ	ቯ				
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ	ቷ				
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ	ቿ				
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ	ኗ				
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ	ኟ				
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ	ኧ				
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኲ	ኳ	ኴ	
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ	ዀ	ዂ	ዃ	ዄ	
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ	ዟ				

ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ	ዧ				
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	ዷ				
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ	ጇ				
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ	ጧ				
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ	ጯ				
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ	ጿ				
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ	ፏ				
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ	ፗ				

Appendix B: Sample data before annotation

Original script	Transliterated
ዘወረ	zäwärä
ዘወረት	zäwärätə
ዘወረኒ	zäwäräni
ዘወረኪ	zäwäräki
ዘዊሩ	zäwiru
ዘዊሩዋ	zäwiruwa
ዘዊሩኪ	zäwiruki
ዘዊሩክነ	zäwirukənə
ዘዊሩና	zäwiruna
ዘዊሩኩም	zäwirukumə
ዘዊሩዋነ	zäwiruwänə
ዘሪጉኪ	zäriguki
ዘሪጉካ	zäriguka
ዘሪጉና	zäriguna
ዘሪጉኩም	zärigukumə
ዘሪጉክነ	zärigukənə
ዘሪጉዋ	zäriguwa
ዘሪጉዋነ	zäriguwänə
ዘሪጋ	zäriga
ዘሪጋቶ	zärigato
ዘሪጋተነ	zärigatänə
ዘሪጋትኒ	zärigatəni
ዘሪጋቶም	zärigatomə
ዘሪጋትኩም	zärigatəkumə

Appendix C: Sample annotated data

ዘወረ[S]

ዘወረ[S]ት[A]

ዘወረ[S]ኒ[E]

ዘወረ[S]ኪ[H]

ዘዊሩ[S]

ዘዊሩ[S]ዋ[O]

ዘዊሩ[S]ኪ[H]

ዘዊሩ[S]ክን[K]

ዘዊሩ[S]ና[G]

ዘዊሩ[S]ኩም[L]

ዘዊሩ[S]ወን[T]

ዘሪጉ[S]ኪ[H]

ዘሪጉ[S]ካ[J]

ዘሪጉ[S]ና[G]

ዘሪጉ[S]ኩም[L]

ዘሪጉ[S]ክን[K]

ዘሪጉ[S]ዋ[O]

ዘሪጉ[S]ወን[T]

ዘሪጋ[S]

ዘሪጋ[S]ቶ[Q]

ዘሪጋ[S]ተን[T]

ዘሪጋ[S]ትኒ[E]

ዘሪጋ[S]ቶም[U]

ዘሪጋ[S]ትኩም[L]

zäwärä[S]

zäwärä[S]tə[A]

zäwärä[S]ni[E]

zäwärä[S]ki[H]

zäwiru[S]

zäwiru[S]wa[O]

zäwiru[S]ki[H]

zäwiru[S]kənə[K]

zäwiru[S]na[G]

zäwiru[S]kumə[L]

zäwiru[S]wänə[T]

zärigu[S]ki[H]

zärigu[S]ka[J]

zärigu[S]na[G]

zärigu[S]kumə[L]

zärigu[S]kənə[K]

zärigu[S]wa[O]

zärigu[S]wänə[T]

zäriga[S]

zäriga[S]to[Q]

zäriga[S]tänə[T]

zäriga[S]təni[E]

zäriga[S]tomə[U]

zäriga[S]təkumə[L]

Appendix D: Sample list of instances

y,ə,r,u,w,ä,n,ə,=,=,=,=,=,0
ə,r,u,w,ä,n,ə,=,=,=,=,=,=,T
=,=,=,=,=,=,z,ä,y,ə,r,a,t,0
=,=,=,=,=,=,z,ä,y,ə,r,a,t,a,0
=,=,=,=,=,z,ä,y,ə,r,a,t,a,=,0
=,=,=,=,z,ä,y,ə,r,a,t,a,=,=,0
=,=,=,z,ä,y,ə,r,a,t,a,=,=,=,0
=,=,z,ä,y,ə,r,a,t,a,=,=,=,=,S
=,z,ä,y,ə,r,a,t,a,=,=,=,=,=,0
z,ä,y,ə,r,a,t,a,=,=,=,=,=,=,O
ä,y,ə,r,a,t,a,=,=,=,=,=,=,=,0
=,=,=,=,=,=,z,ä,y,ə,r,a,t,0
=,=,=,=,=,=,z,ä,y,ə,r,a,t,ə,0
=,=,=,=,=,z,ä,y,ə,r,a,t,ə,n,0
=,=,=,=,z,ä,y,ə,r,a,t,ə,n,i,0
=,=,=,z,ä,y,ə,r,a,t,ə,n,i,=,0
=,=,z,ä,y,ə,r,a,t,ə,n,i,=,=,S
=,z,ä,y,ə,r,a,t,ə,n,i,=,=,=,0
z,ä,y,ə,r,a,t,ə,n,i,=,=,=,=,0
ä,y,ə,r,a,t,ə,n,i,=,=,=,=,=,0
y,ə,r,a,t,ə,n,i,=,=,=,=,=,=,0
ə,r,a,t,ə,n,i,=,=,=,=,=,=,=,E
=,=,=,=,=,=,=,z,ä,y,ə,r,a,t,0
=,=,=,=,=,=,z,ä,y,ə,r,a,t,ə,0

Appendix E: sample testing process and result from TiMBL

```
Starting to test, Testfile: largtest.test
Writing output in:        largtest.test.IB1.O.nw.k1.ID.out
Algorithm     : IB1
Global metric : Overlap
Deviant Feature Metrics:(none)
Weighting     : No Weighting
Decay         : Inverse Distance
Tested:     1 @ Tue Jul 31 17:31:52 2018
Tested:     2 @ Tue Jul 31 17:31:52 2018
Tested:     3 @ Tue Jul 31 17:31:52 2018
Tested:     4 @ Tue Jul 31 17:31:52 2018
Tested:     5 @ Tue Jul 31 17:31:52 2018
Tested:     6 @ Tue Jul 31 17:31:52 2018
Tested:     7 @ Tue Jul 31 17:31:52 2018
Tested:     8 @ Tue Jul 31 17:31:52 2018
Tested:     9 @ Tue Jul 31 17:31:52 2018
Tested:    10 @ Tue Jul 31 17:31:52 2018
Tested:   100 @ Tue Jul 31 17:31:52 2018
Tested:  1000 @ Tue Jul 31 17:31:52 2018
Ready:    1328 @ Tue Jul 31 17:31:52 2018
Seconds taken: 0.1313 (10117.02 p/s)

overall accuracy:       0.915663  (1216/1328),
```