ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA

AUTOMATIC MORPHOLOGICAL ANALYZER FOR AMHARIC
AN EXPERIMENT EMPLOYING UNSUPERVISED LEARNING AND AUTOSEGMENTAL ANALYSIS APPROACHES

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE**

BY TESFAYE BAYU BATI
JUNE, 2002

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA

AUTOMATIC MORPHOLOGICAL ANALYZER FOR AMHARIC
AN EXPERIMENT EMPLOYING UNSUPERVISED LEARNING AND AUTOSEGMENTAL ANALYSIS APPROACHES

BY
TESFAYE BAYU BATI

Signature of the Board of Examiners for Approval

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Declaration

This thesis is my original work and has not been submitted as a partial requirement for a degree in any university

_____

Tesfaye Bayu Bati
June 2002

The thesis has been submitted for examination with our approval as university advisors.

1. Ato Tesfaye Biru _____
2. Dr. Zelealem Leyew _____
3. Ato Mesfin Getachew _____

DEDICATION

To my beloved father, Bayou Bati, whose life long effort to help others know what he knows brought me up for this status; and to my elder brother, Limenew Bayou, whose education is discontinued to help me pursue my university education.

ACKNOWLEDGMENT

I am deeply indebted to the critical comments made by my advisors, Ato Tesfaye Biru, Dr. Zelealem Leyew and Ato Mesfin Getachew without whose close follow-up my theses would never have been successful. I also extend my thanks to Ato Million Meshesha and W/ro Woinshet Abdella of SISA for their comments, suggestions and assistance in Visual C++. I am also grateful for the assistance of Ato Daniel Abera of the Linguistics Department, AAU, for reviewing my proposal and helping me evaluate the experimental result.

My special appreciation also goes to Prof. John Goldsmith of the University of Chicago for provision of the source code of his system – Linguistica2001. His comments and advises were also my encouragements and directions to go through so many concepts, ideas and principles completely new for me.

Moreover, I'll never forget the support I got from my goodhearted friends, W/rt Ribka Shanko and Ato Amsalu Gobena of Debub University, in taking wholeheartedly all the responsibilities on behalf of me. I would also like to express my gratitude to Enchalew Yifru, Getaw Shumye and Solomon Mulugeta who were my invisible hands for my stay in Addis.

So many people – classmates, instructors, friends, family members, and colleagues come to my mind with their supports, encouragements, and prayers. May God bless you all!

TABLE OF CONTENTS

TABLE OF CONTENTS ii
LIST OF FIGURES AND TABLES v
ABSTRACT vii
LIST OF ABBREVIATIONS vi

1 INTRODUCTION 1

1.1 BASIC CONCEPTS IN MORPHOLOGY 1
1.1.1 TYPES OF MORPHOLOGICAL PROCESSES 3
1.1.2 CONSTITUTES OF A MORPH 4
1.1.3 THE STRUCTURE OF WORDS (OR MORPHOTACTICS) 6
1.1.4 NONCONCATENATIVE MORPHOLOGY OF SEMITIC LANGUAGES 7
1.2 BACKGROUNDS OF THE STUDY 8
1.3 STATEMENT OF THE PROBLEM AND ITS JUSTIFICATION 11
1.4 OBJECTIVES OF THE STUDY 13
1.4.1 GENERAL OBJECTIVES 13
1.4.2 SPECIFIC OBJECTIVES 13
1.5 METHODS 14
1.5.1 LITERATURE REVIEW 14
1.5.2 PREPARATION OF CORPUSES 14
1.5.3 DEVELOPMENT OF THE PROTOTYPE MORPHOLOGICAL ANALYZER 15
1.5.4 TESTING TECHNIQUES (OR PROCEDURES) 16
1.6 APPLICATIONS OF THE RESULTS 16
1.7 SCOPE OF THE STUDY 17
1.8 ORGANIZATION OF THE THESIS 18

2 COMPUTATIONAL MORPHOLOGY 19

2.1 TASKS IN MORPHOLOGY 19
2.2 APPROACHES TO MORPHOLOGICAL ANALYSIS 20
2.3 UNSUPERVISED LEARNING FOR MORPHOLOGICAL ANALYSIS 23
2.3.1 MINIMUM DESCRIPTION LENGTH (MDL) FRAMEWORK 23
2.3.2 UNSUPERVISED LEARNING ALGORITHMS 25
2.3.2.1 Initial Splitting 25
2.3.2.2 Identification of Signatures 27
2.3.2.3 Optimizing the Morphology using heuristics and MDL 28
2.3.2.4 Triage 33
2.3.2.5 Determining Paradigms 34

2.4 REMAINING ISSUES FOR SEMITIC MORPHOLOGY 34
2.4.1 AUTOSEGMENTAL REPRESENTATION OF SEMITIC MORPHOLOGY 35

3 THE STRUCTURES OF AMHARIC WORD 37

3.1 THE AMHARIC PHONETICS 37
3.2 AMHARIC WORD CLASSES 39
3.3 WORD FORMATION 40
3.3.1 STEM FORMATION 41
3.3.1.1 Gemination 42
3.3.1.2 Reduplication 44
3.3.2 SOME POINTS ON AMHARIC STEM TEMPLATES 46
3.3.3 AMHARIC INFLECTIONS (AFFIXATION) 47

4 CORPUS PREPARATION AND ALGORITHM DESIGN 49

4.1 CORPUS PREPARATION 49
4.2 ALGORITHM DESIGN 52
4.3 DATA STRUCTURES 60

5 THE EXPERIMENT 63

5.1 EXPERIMENT WITH LINGUISTICA2001 63
5.1.1 THE TEST/EXPERIMENT ENVIRONMENT 64
5.1.2 INITIAL INSPECTION OF AMHARIC SIGNATURES 65
5.1.3 THE TEST WITH LINGUISTICA2001 71
5.2 EXPERIMENT WITH THE PROTOTYPE STEM ANALYZER 73
5.2.1 THE STEM SIGNATURE 74
5.2.2 THE TEST WITH ASMA 76

6 CONCLUSION AND RECOMMENDATIONS 79

6.1 CONCLUSION 79
6.2 RECOMMENDATIONS 82

7 REFERENCE 85

8 APPENDIX.....1

Appendix 1: Character representation used in the Transcription..... 1
Appendix 2: A Screen Snapshot of ASMA 1
Appendix 3: A Corpus used for Expeiment with Linguistica2001 2
Appendix 4: A Corpus used in the experiment with ASMA..... 2
Appendix 5: Partial Visual C++ Source Code of ASMA..... 2

LIST OF FIGURES AND TABLES

Tables

Table 3-1Amharic Consonants 37
Table 4-1: Changes made in phonemic representation of Amharic alphabets 50
Table 5-1: Thresholds Used in the experiment 65
Table 5-2: Prefix Signature Identified 66
Table 5-3: Suffix Signatures identified in the first run 67
Table 5-4: Amharic Signatures Identified 70
Table 5-5: Test Result 73
Table 5-6: Result of Stem Component Analysis 77

Figures

Figure 2-1: **Example of autosegmental representation in phonology** 35
Figure 2-2: An example of Autosegmental Representation 36
Figure 4-1: An algorithm for Autosegmental representation of Amharic Stems 53
Figure 4-2: An algorithm to read a word from a file 54
Figure 4-3: Algorithm for extracting morphemic component of stems 54
Figure 4-4: Algorithm for extracting morphemic components of stems (modified) 57
Figure 4-5: Algorithm for Constructing Autosegmental Tiers 57
Figure 4-6: An algorithm for creating an association between autosegmental tiers 58
Figure 4-7: An algorithm to add addresses of morphemes to a Stem Signature 59
Figure 4-8: An Algorithm to search autosegmental tires 59
Figure 4-9: Data Structure of the Autosegmental Tiers 61
Figure 4-10: Data Structure for the Stem Signature 61
Figure 4-11: Data structure to represent morpheme associations 62
Figure 5-1: An example of Stems appearing with a particular signature 68
Figure 5-2: Sample Stem Signature 75

LIST OF ABBREVIATIONS

Code	Description
/ /	Word form
[ ]	phone/phoneme
{ }	morpheme
‘ ’	gloss
+	Morpheme Boundary
1.	First Person
2.	Second Person
3.	Third Person
acc.	Accusative
adju.	adjutative
c:	Common
def.	definite article
f:	feminine
g.	Gerundive
Imp.	Imperfect
intr:	Intransitive
J.	Jussive
m.	masculine
NULL	designate word end
O	Object
Per.	Perfective
pl.	plural
POS	Possessive
rcp	reciprocal
sg.	Singular
S	Subject
sth.	Something
tr.	Transitive
cs-rcp	causative reciprocal
* (as in *unsad)	ill-formed structure
UL	Unsupervised Learning
TLM	Two Level Model of Morphology

ABSTRACT

Automatic understanding of natural languages requires a set of language processing tools. A ***morphological analyzer,*** *which parses words into their morphemic components, is one of these tools.* *This thesis reports an attempt intended to develop such a tool for* ***Amharic***.

Word formation in Amharic involves three levels of morphological operations – ***stem formation***, ***affixation*** *and* ***cliticization***. *Since affixation and cliticization are similar with those in Indio-European languages, a language independent system tested in these languages is used. The system, called* ***Linguistica2001***, *creates morphological dictionary (called signature) by extracting prefixes, stems and suffixes from a given corpus. The system uses the modified version of* ***Harris's Algorithm of Successor Frequency*** *to detect plausible word break points. Additional heuristics are used to improve the word breaks produced.* ***Minimum Description Length (MDL)*** *test serves as a benchmark to accept a signature as part of the morphology of a given language.*

For the stem internal operations, another approach based on ***the principle of autosegmental Phonology*** *is used. This principle represents phonemic features of a word in* ***different tiers*** *and uses* ***association lines*** *to maintain their relationships. This approach is used to design algorithms and data structures required for extraction and representation of stem components. A prototype system, called* ***Amharic Stems Morphological Analyzer (ASMA),*** *is developed to test the algorithms. Though the two systems are tested separately, ASMA is designed to work in an integrated manner by accepting as its input stems identified by Linguistica2001.*

The experiment is conducted using ***corpuses prepared in this study***. *The experimental result obtained is encouraging. Linguistica2001 parses successfully 87% of words of the test data (433 of 500 words). This result corresponds to a precision of 95% and a recall of 90%. The second system analyses 241 (or 94%) of the255 sample stems correctly.*

CHAPTER I

Introduction

This chapter has two sections. The first section gives brief introduction to concepts in morphology relevant to the current work. The second section describes the problem area and the general framework of the study. For detailed information on the issues discussed, readers are advised to consult works listed in the references or to read any introductory book in Linguistics.

1.1 Basic Concepts in Morphology

Linguistics is concerned with languages and their structures. It studies languages at different levels such as at phone, word, sentence and the like. There are also different branches in Linguistics to deal with specific features of a language. The four major branches of linguistics commonly available in the literature are phonology, morphology, syntax and semantics (Hundson, 1999).

Phonology concerns sounds (or phones) and their features[1]. The general understanding in linguistics is that human utterance is produced from distinct sounds (or phones) that in combination are used to form words or any other meaningful linguistic entity (Hundson, 1999). Hence, Phonology also tries to identify rules that govern combination and co-occurrences of phones in words and sentences. Morphology, on the other hand, concerns words and their internal structures (Goldsmith, 2001a). It deals with constituents of words and the rules that govern their co-occurrences in words. At the next higher level, Syntax concerns the combinations of words as phrases and phrases as sentences and the rules that govern the

[1] Sounds have different features depending on their point of articulation, whether they are vowel or consonant and the like

combinations. Semantics is concerned with word and sentence meanings and their interpretations.

These four areas of linguistics are interrelated. Combined together, they are used to describe the linguistic nature of a particular language. Of the four, this paper concentrates on morphology, principal concern of which is words and their internal structures. The basic assumption behind morphology is that the infinity of words of a language are produced from a finite collection of smaller units (Trost, 2000).

These smaller units which are used to from words are called ***morphemes***. For example if we see /helped/ in a sentence, we know that it is produced from {help} and {-ed}[2], each of which are a separate morpheme. Each component has also semantic or grammatical information to add to the overall meaning of the word. For example, {help} carries the semantic information and {-ed} tells the grammar (or more specifically, the tense) of the word. However, the two morphemes must be combined in some way to form the word /helped/.

The process involved in forming words from one or more morphemes is called ***word formation.*** In word formation, a morpheme can make some changes or can have a number of ways to form the word. For example, the morpheme {*door*} forms the word /*door/* without any change. On the other hand, if you take a plural in English, it is represented in a number of ways such as {-*s*} as in /dogs/, through a morph {-*en*} as in /*oxen/*, through stem vowel alteration as in /men/ and /women/ or through zero morph[3] as in /sheep/. Such a process of making a morpheme a word (or word components) is called ***morpheme realization.*** Moreover, different forms of morpheme realizations are called a ***morph*** while all different forms used to

[2] The hyphen (-) in the bound morpheme {-ed} shows the position where fixation is to be made.
[3] The plural of 'sheep' is represented as {sheep} + {} or a sheep is said to have a null plural morph.

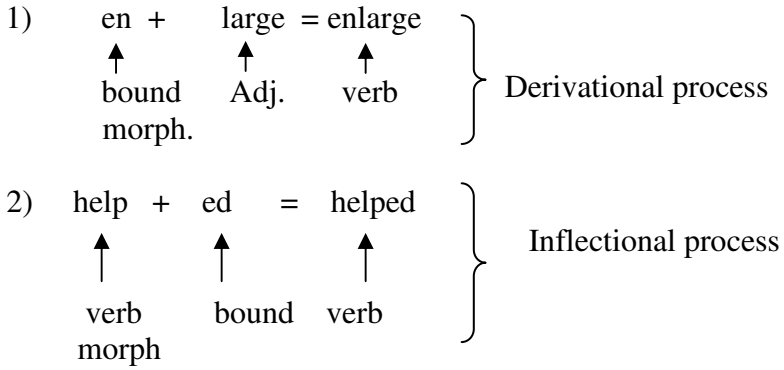
represent a morph are called ***allomorphs.*** For example, the different forms of representing plurals in English can be referred as allomorph of a plural morph.

Other related concepts are that of ***free/bound*** morphemes. A free morph may form a word by its own while bound morphs occur only in combination with other forms. For instance, the morph {door} stands as a word /door/ by itself; so it is a free morph. But {-s} can not stand alone as an independent word; hence it is a bound morph. In word forms such as /doors/, for example, there are two morphemes the free morph {door} and the bound morph {-s}, of which the morph {-s} cannot stand by itself.

1.1.1 Types of morphological Processes

Morphological processes can be categorized into inflectional and derivational. The two processes differ in the type of words they produce. Derivational processes create new words of different word class[4] (for example generating a noun from a verb root). While inflectional process does not change the word class of the word created. The following are some examples: -

[4] Refers to roles word plays in a sentence. For example nouns serve as subject or object in a sentence.



In the first example, the adjective {large} is changed to a verb {enlarge} by attaching a bound morph {en-}. This change from adjective to verb shows that the morphological process involved is derivational; while in the second case, because both /help/ and /helped/ are verbs, the process involved in forming /helped/ from {help} and {-ed} is inflectional.

1.1.2 Constitutes of a morph

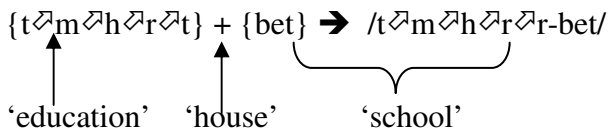
We have seen that words are formed from a combination of morphs. Word forms[5] are produced from morphs in a number of ways. Trost (2000) presents ***affixation***, ***reduplication*** and ***compounding*** as the major ones.

Affixation attaches ***affixes*** to free morphs (called stems) to form words. An affix is a bound morph that is realized as a sequence of phonemes (or graphemes). Affixes can be ***suffixes***, ***prefixes***, ***infixes***, or ***circumfixes***. A suffix is an affix that is attached after a free stem whereas a prefix is an affix that is attached in front of a stem. The plural maker {*-s} and {-ed}* are examples of suffixes. In word formation, they are always attached to the right of stems as in /dogs/ and /worked/. On contrary, {*un-},{en-}, {in-},* as in *uncommon, enlarge, indifference,* are examples of prefixes and they are attached to the left of a stem.

[5] Word forms refers to all words produced from a given word by morphological process (such help, helps, helping, helped of the main word {help})

Suffixes and prefixes are the two most common affixes found in many languages. But some languages have also circumfixes and infixes (Trost, 2000). A ***crucifix*** is the combination of a prefix and a suffix which together express some feature. Trost (2000) gives examples from the German past participle markers {ge…t} and {ge…n} as in /gesagt/ '*said'* of /sagen/ and /gelaufen/ '*run'* of /laufen/. In Amharic the 3f.sg. imperfect marker {t⬀…*äčč} as in /t⬀-mät-all-äčč/ 'she'll come'* seem an example. An ***infix***[6] is affix where the placement is defined in terms of some phonological condition(s).

Compounding is similar with affixation. Their main difference lies in the type of morphs they combine (Trost, 2000). Affixation attaches a bound morph onto a free morph whereas compounding combine two freely standing morph to form another word forms. The following is an example of compounding in Amharic: -



Morph realization in ***reduplication*** differs from the two discussed above. Reduplication copies some portion of a free morph to form another word form (Hudson, 1999). An example of reduplication in Amharic is presented in Trost (2000) as follows: -

[6] see the discussion on reduplication for examples

/kättäfä/ 'he chopped' → {kä-ta-ttäfa} /kätattäfa/ 'chop again and again'
/k'ännäsä/ 'he decreased'→ {k'ä-na-nnäsä} / k'änannäsä/ 'decrease a lot'

The examples show that /kätattäfa/ *and* /k'änannäsä/ *are* produced from their base words by reduplicating the second consonants (or penult) i.e., [t] of /kättäfä/ and [n] of /kännäsä/ and infixing the vowel [a].

To summarize, the forgoing discussion presented three ways of morph realization – affixation, compounding and reduplication. But there are also other factors involved in morph realization (Trost, 2000). These factors are discussed in the following section.

1.1.3 The structure of words (or Morphotactics)

Languages have patterns or structures in which morphs are combined to form words. These word patterns (or word structures) are called morphotactics. For example, the English word *pseudohospitalization* is formed from /pseudo-/, /hospital/ /-ize/ and /-ation/. But these morphemes can be randomly arranged as *hospitalationizepseudo, *pseudoizehospitalation, *pseudohospitalationize if such a morphotactics exits in the language.

Languages have syntactic, phonological, semantic or purely lexical constraints that prohibit construction of such ill-formed word forms (Trost, 2000). For example, a semantic restriction is seen on the English negation prefix {*un-}*. The negative meaning it has prevents its attachment to words that already have a negative meaning (for instance it is possible to say 'unhappy', but not *unsad). Phonological restriction is observed in English prefix ending with [n]. As shown in the examples below, [n] of the prefix {in-} is changed to [m] when it is attached to stems with labial[7] initials.

[7] Labials are phones that are articulated by stopping the outgoing air by lips.

{in-} + {feasible} → /infeasible/
{in-} + {mature} → /immature/
{in-} + {proper} → /improper/

In the example above, [m] of /mature/ and [p] of /proper/ are both labial consonants. As a result [n] of the prefix {in-} is changed to [m]. We can see that [n] is not changed to [m] in /infeasible/ because [f] is not labial (Trost, 2000).

1.1.4 Nonconcatenative Morphology of Semitic Languages

The preceding sections presented general concepts in Morphology. This subsection focuses on peculiar feature of Semitic languages[8] – a family of languages spoken in Middle East and North and North East Africa (Bender and Hailu, 1978). The peculiar feature of Semitic morphology lies in the way word stems are produced.

In English and other Indio-European languages, a stem is a free morph that cannot be further decomposed into meaningful parts (McCarthy, 1981). However, this is not the case for Semitic languages. A Semitic stem is produced from three morphemes - consonantal roots (referred as root hereafter), vocalic patterns and stem templates. The root morpheme is believed to carry the semantic information whereas the vocalic melodies together with the template are used to decide the grammatical category of the stem (Trost, 2000). The following are examples of stems produced from Arabic root {ktb} 'write' (adopted form McCarthy, 1981):

Stem	**Gloss**
kataba	'he wrote'
kattaba	'he caused to write'
takaatabuu	'they kept up a correspondence'
ktataba	'he wrote, copied'
kitaabun	'book'

[8] Examples of languages in this family are Arabic, Hebrew, Amharic, Tigrinya, Gurgagna

The stems in the example above include {ktb} as their components. They are also thought to have vocalic patterns such as {a} of /kattaba/, /kaataba/ and /ktataba/, and {au} of /takaatabuu/ and /kitaabun/ and templates such as {CVCVCV}[9] of /kataba/ and {CVCCVCV} of /kattaba/. Moreover, their glosses have meaning related to 'to write'. This similarity in meanings is assumed to come from the common root {ktb} they have. The grammatical variation existed in the stems (for instance /kataba/ is a perfective[10], /kitaabun/ is a noun) is thought to be due to the vocalic melodies and templates used. For example, the template {CVCVCV} together with the vocalic melody {a} tells that the stem /kataba/ is a perfective (McCarthy, 1981).

In stem formation, the consonantal roots and vocalic melodies are combined in a non-linear way following the CV-pattern depicted in the template (Trost, 2000). As an example /kataba/ is formed by infixing {a} into {ktb} following the {CVCVCV} template (i.e., by replacing C by elements form the root and V by vocalic melody elements). As a result, it is difficult to get a morpheme from Semitic stem by searching the stem linearly for a morphemic component; rather this is done by categorizing the stem components into two parts checking whether a character at specific point in the stem is a consonant or a vowel. This type of morpheme is called ***discontinuous morpheme***, and the morphology with such morpheme types ***nonconcatenative morphology*** (McCarthy, 1981).

1.2 Backgrounds of the Study

Allen (1995) stated that most of human knowledge is recorded in linguistic form, i.e., in the form of natural language[11] (NL) texts and utterances. This reliance on NL makes understanding of natural language crucial for improved knowledge representation. Review of previous researches in the area of natural language also proves this fact (Warner, 1987).

[9] C stands to 'Consonant' and V to 'Vowel'.
[10] Refers to an action already occurred.
[11] The term 'natural language' is used to refer to languages in which we speak, write or communicate.

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, 1995). 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) label 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 (Allen, 1995). Other system such as part of speech taggers and sentence parsers are developed for higher-level linguistic processing such as syntactic and semantic analyses.

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 (Goldsmith, 2001c). 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 (Goldsmith, 2001c). 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.

However, development of computerized morphological systems for a language is not a trivial task. It requires identification and proper coding of the word-formation principles in the form of computer programs (or algorithms). This is a complex task. Trost (2000) indicates that the way morphemes are produced from phones, the phonological feature of sounds in a morpheme; the writing system and hyphenation style of the language and existence of irregular style of word formation pose difficulty in the development of a computerized system for a language.

Moreover, languages vary in their word formation structure. We can see such difference considering the case of Amharic, English and Finish, for example. An English stem, for instance, can not be further subdivided in to meaningful parts whereas Amharic stems do. On the other hand, Finish concatenates morphemes one onto the other without any change during word formation whereas English and Amharic inflict one of the morphemes to fit phonologically to the other (Trost, 2000 and Hundson, 1999).

As a result, due consideration of the phonological, morphological, and orthographic (writing system) features of a language is essential to develop a morphological system to a particular language. Moreover, the algorithm developed needs to be computationally efficient (in memory, and processing speed). Such complexities involved in word formation make development of morphological systems a non-trivial task. It requires thorough study of the specific linguistic features of a language, and developing an algorithm which is effective and efficient computationally.

1.3 Statement of the Problem and its Justification

According to the latest census result, Amharic is a mother tongue of more that 17 million people. The language is also used as a second language for over 5 millions people (ECSA[12], 1998). It has also been, for a long period, the principal literal language and medium of instruction and school subject in primary and secondary schools of the country. Moreover, it is the official working language of the Ethiopian Federal Government, all of which make the language to be predominantly used in word processing activities in different offices. Furthermore, there are also a large number of documents, published and unpublished, written and recorded in Amharic.

From review of the number of works done on the linguistics aspect of the Amharic language, it seems that the language is studied intensively (Baye (1999), Leslau (1995), Bender and Hailu (1978), for example). However, corresponding development in the computational aspect is very rare. Though there are few research works initiated to address the problem, the majority of these researches are carried out in the area of character recognition (OCR) (Worku, 1997; Dereje, 1999; Ermias, 2000; Million, 2000) and development of Amharic text retrieval systems (Birru (1992), Nega (1999), and Saba (2001)). There are also some works in the area of speech to text recognition and text to speech synthesis (Solomon, 2001 and Laine, 1998).

In the area of NLU system development, however, the only works I found are Abiyot (2000) and Mesfin (2001). Mesfin (2001) addressed the problem of developing an automatic part-of-speech (POS) tagging system and Abiyot (2000) attempted to develop a morphological analysis system for verbs and nouns derived from verbs.

[12] ***ECSA*** stands for Central Statistical Authority of the Federal Democratic Republic of Ethiopia

Automatic POS tagging system, as shown in Mesfin (2001), is a task which assigns a syntactical category or part of speech tags to words in sentences. However, lack of Amharic morphological analyzer had an impact on the quality of the output the prototype tagger produced. Mesfin has indicated that his tagging system does not give inflectional categories of words (such as gender, number, tense, case, aspect and the like).

Apart from POS tagger, there is also a need for morphological analyzers in syntactic and semantic parsers (Antworth, 1994). Amharic words, especially verbs, not only inflict for a number of grammatical cases but they can also add a number of clitics such as object pronouns, independent pronouns, and possessive pronoun clitics (Mullen, 1986). For example, the verb /asfärrädäčč⇗b⇗ňň/ is a single word with a meaning expressible by a full sentence. It is equivalent with the English sentence *'a case she intitated against me was decided in her favor.'* The word is built from the causative prefix {as-}'causes', a perfective stem {färrädä}'judged', a 3sg.f. subject maker clitics {-äčč} 'she', a benficative marker {b}'against' and the object pronoun {-ňň} 'I'. Mullen (1986) shows that syntactic and semantic parsing of sentences with such words is difficult without first breaking the words into their constituent morphemic components. Morphological analyzers are thus required to handle such preprocessing for the higher-level syntactic and semantic parsers.

In this line, Abiyot (2000) has attempted to develop a word parsing system. The prototype system he developed uses different dictionaries (for suffix, prefix, and stem) and hand-encoded rules (represented as algorithms) to provide that information. The experimental result of the system based on 200 verbs and 200 nouns shows a 94% accuracy level in morphological parsing.

However, I have noted some limitations in Abiyot's work. Firstly, the word parser developed is meant only for verbs and nouns derived from verbal roots. Thus, further work needs to be done to make his system complete by developing components for the remaining word classes. Moreover, his system is a rule-based system, that is, the system is dependent on word formation rules encoded as an algorithm. Such full reliance on rules and stored dictionary seems to limit the adaptability and applicability of the system to words and word forms not incorporated in the dictionary (also called unknown words).

Therefore, further investigation in the area of morphological analysis system development was felt worth considering. In addition, a study with other approaches like stochastic (or probabilistic approaches) opens an alternative ways of carrying out the task. The current study is, thus, basically initiated to meet these two ends - filling the existing gaps in Abiyot work and exploring the possibility of employing stochastic approaches in morphological analysis.

1.4 OBJECTIVES OF THE STUDY

1.4.1 General Objectives

The general objective of this study is to assess the possibility to develop an automatic (computer-based) morphological analysis system for Amharic language.

1.4.2 Specific Objectives

To achieve the general objective, the study has attempted to address the following specific objectives: -

1. to understand phonological, morphological and orthographic features of Amharic and the phonological and morphological processes involved in Amharic word formations and conjugations.

2. to assess different techniques and approaches employed so far in morphological analysis tasks and select the ones that seem appropriate to the morphological property of Amharic.
3. to organize training and test corpus data
4. to adopt or develop a machine learning algorithm
5. to adopt or develop a morphological analysis system that employs the machine learning algorithm mentioned above.
6. to train the morphological system adopted/developed with training corpus data
7. to test the effectiveness and appropriateness of the morphological analysis technique developed.
8. to discuss and report the experimental results found

1.5 METHODS

1.5.1 Literature Review

Research reports, books, journal articles and other published and unpublished documents (including those from Internet) are consulted. Literature review is conducted for various reasons. Related works in computational morphology are reviewed to identify different approaches being tested in development of morphological analysis systems; to examine and select appropriate machine learning algorithm; and to know how to develop corpus data for morphological analysis research work. Works in Amharic linguistics are also reviewed to understand the morphological structure of Amharic words.

1.5.2 Preparation of corpuses

In this study an approach that employs an unsupervised learning mechanism is selected to develop the morphological analysis system. This approach requires a large corpus data, the corpus being electronic text data consisting of list of words such as those found in newspapers and books. A corpus consisting some 5, 236 words is prepared as part of this research.

Furthermore, due to a problem encountered during the experiment (discussed in chapter 5), another set of corpus consisting of some 300 stems are collected from articles and books on Amharic linguistics.

1.5.3 Development of the prototype morphological analyzer

Two separate systems are used to develop the morphological analysis system. The first system, Linguistica2001 that is developed by Goldsmith (2001a), is used to identify the prefix, stem and suffix components of words in the corpus. Though this system could not handle the nonconcatenative morphology of Semitic stems, it seems applicable to handle two of the three major morphological processes involved in Amharic word formation – infliction and cliticization (See Chapter 3).

For this reason, the system is used as a component and another prototype system is developed to handle the stem internal morphological analysis tasks. The prototype system, which is labeled ***Amharic Stems Morphological Analyzer (ASMA)***, is developed using Visual C++. Different data structures and algorithms (searching, insertion, morpheme component identification etc.) that are required for the system are also designed as part of this study. Considering the intensive searching and insertions activities involved, the data structure designed are binary-search trees.

Though the two systems are not integrated in this study due to time limitation, the prototype system developed can use the output of Linguistica2001 as an input to finalize the analysis task of Amharic words. Integration is also one of the reasons to use Visual C++, using which Linguistica2001 is developed, in the development of the prototype system.

1.5.4 Testing Techniques (or Procedures)

In the study, unsupervised learning approach is used to develop the morphological analyzer. The system developed depends purely on patterns within words of the corpus to identify morphemic components of words. The quality of the outputs found is then tested by examining whether the systems produce linguistically acceptable morphemic components such as prefixes, suffixes, and stem components.

For this purpose, 500 words of the first corpus and 255 of the second corpus are selected as ***test data*** and given along their word breaks (outputs of the systems) to 2 linguists (instructors at AAU) to evaluate the correctness of the result. The test data are selected by taking the specified number of words from the alphabetized lists of words in the corpus. The performance of the systems is tested based on the evaluation given by the linguists. The system's performance is presented in ***percentage***, and ***precision*** and ***recall*** ratios. The precision and recall values are computed using the formula used in Goldsmith (2001a). The two measures can be used to measure the accuracy rate of the system in terms of the quality of the analysis (precision) and the exhaustively of the analysis given (recall) which is similar with the notion they have in information retrieval (Salton, 1983).

1.6 Applications of the Results

A morphological analyzer is one of the components used to develop NLU systems. Integrated in a full-fledged NLU system, it can be used in a number of application areas such as in full text search and information retrieval, in automatic document highlighting and summarization, in machine translation systems and in fuzzy word matching systems used in Optical Character

Recognition (OCR). A good description of the application areas is found at Xerox web site at http://www.xrce.xerox.com and in Abiyot (2000).

1.7 Scope of the Study

Amharic words have complex morphological structure caused by a number of phonological, morphological and syntactic processes involved. Development of a morphological system for the language thus requires components that handle each processes involved. Because of limited time available for the project, the analysis system tested and developed considered only morphological processes involved – identification of affixes and extractions of stem components. For the same reason, the size of the corpuses prepared and used in this study is the least size recommended.

Moreover, morphological analysis task using the Goldsmith's Linguistica2001 system involves many tasks such as initial segmentation, forming signature, Optimization with Heuristics and MDL (discussed in Chapter 2), employing triage and paradigmatic analysis. In this study, though an attempt was made, the MDL test and the paradigmatic analysis were not performed. This is due the following reasons. Firstly, I found the MDL module of Linguistica2001 inconsistent (communicating with the developer, I found that the MDL module is still under development).

The paradigm test is conducted by evaluating pattern of affix uses. However, since an affix can be used for a number of grammatical cases, the module requires large size corpus to identify proper paradigms. But the corpus used in this study is too small to get such results.

1.8 Organization of the Thesis

This thesis is organized in six chapters. Chapter one introduces basic concepts in morphology considered important to understand issues rose in the paper. The chapter also presents the objective and statement of the problem of the study. Different approaches for developing morphological analyzers are described in Chapter 2 with a special emphasis on the one to be used in this study. The third chapter describes the morphology of Amharic words and the word formation and conjugation processes. Chapter 4 describes the corpus, algorithms and data structures designed for the study. The experiment conducted and results obtained are discussed in chapter five. The conclusions and recommendations made in the study are presented in chapter six. Finally, the corpuses used and the source code of the prototype system developed are attached as appendixes.

CHAPTER II

Computational Morphology

This chapter presents review of literature on computational morphology. The chapter begins with a brief introduction to the two major processes in computational morphology - recognition and generation. Presenting further discussion on various approaches being tested in word form recognition, the chapter gives detailed account on unsupervised learning approach employed in the development of Linguistica2001. The chapter ends up with a description on an approach used to handle morphological processes internal to Semitic stems.

2.1 Tasks in Morphology

As a subfield of linguistics, morphology deals with internal structure of words. In this respect, computational morphology is intended to handle this task automatically with the use of computers and computational methods. Basically, the task involved in computational morphology can be grouped in to two (Antworth, 1994): -

a) Word-form recognition and generation
b) Part of speech (POS) or inflectional category determination

Word-form recognition and generation are tasks involved in *tokenizing word forms* into their constitute morpheme components; and *producing word-forms* from their ingredient morphemes respectively. Word form generator accepts as input a lexical form (such as *spy + s*) and returns the surface form *spies*. The recognizer performs the reverse, that is, it accepts as input a surface form such as *spies* and returns an underlying form divided into morphemes, namely, *spy + s*. These processes demand identification of word form components (stem and suffixes) and taking account of the regular phonological or orthographical alternations due to morphological, and morphophonological processes involved (Antworth, 1994).

The second process in computational morphology is POS determination. The inflectional category of words is often taken from a morpheme that serves as a "head of a word[13]". For example, the English word *enlargements* come from the following morphemes: /*en-*/ the verb maker prefix; the adjective stem /*large*/; the noun maker suffix /*-ment/* and the plural maker /*-s/*. Of these morphemes, the noun maker suffix /*-ment/* is the head of the word *enlargements,* and thus, the entire word is a plural noun (Williams, 1981).

2.2 Approaches to Morphological Analysis

There are a number of approaches employed in computational morphology. As discussed in Kazakov and Munandhar (2000), some of these approaches are based on concepts in automata theory, probability, principle of analogy, and information theory. Kazakov and Munandhar (2000) broadly categorize these categories into *rule-based* and *corpus-based* approaches.

A rule-based approach is based on a theory of morphology laid down by an expert. Kazakov and Munandhar (2000) stated that 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 (Karttunen, 1994). A good description of rule-based computational morphology is found in Anderson (1988).

The most commonly cited rule-based method is the Two-Level-morphology (TLM) (Koskennieme, 1983). TLM is devised to handles 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

[13] The notion of "head of a word" is described and illustrated in William (1981).

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 (Kiraz, 1995 and Beesley, 2000).

Unlike rule-based approaches, corpus-based approaches do not strictly follow explicit theory of linguistics (Kazakov and Munandhar, 2000). 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 (Kazakov and Munandhar, 2000).

Based on the type of text corpora used, corpus-based approaches can be further categorized into *supervised* and *unsupervised* approaches. 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 (Kazakov and Munandhar, 2000). The work of Nagamatsu and Tanaka (1999) on Japanese morphology is an example. They used morphologically annotated dictionary to train their k-NN-based[14] system. The system employs n-gram[15] data to determine the best point of morphological segmentation of sentences. Similar works by Janssen (1992), Klenk (1992) and Flenner (1994, 1995) are reported in Goldsmith (2001a).

[14] K-NN (Nearest Neghebor) method – is a searching method for similar examples. It uses example database and search for similar pattern in the data.
[15] n-gram – is a string of n, usually adjacent characters extracted from a section of continuous text (Robertson, Willett, 1998).

The unsupervised approaches, on the other hand, do not need such preprocessing on the corpora. Some heuristics or probabilistic information generated from the corpus is used to develop the morphological analysis system (Kazakov and Munandhar, 2000). There are some practical and theoretical factors that make unsupervised learning preferable. Kazakov and Munandhar confirmed that using annotated corpora greatly facilitates learning. However, there are situations in which one is interested in unsupervised learning. The motivation for UL discussed varies from purely pragmatic, such as the high cost or unavailability of annotated corpora, to theoretical, when language is modeled as yet another communication code with in the framework of Information Theory.

Kazakov and Munandhar considered unsupervised learning methods using a combination of genetic algorithm and inductive logic programming techniques. They reported that the hybrid approach is an efficient combination of unsupervised learning to word segmentation. Similarly, Goldsmith (2001a) tested unsupervised learning approaches for morphological analysis and reported a success with a precision rate of 82.9% in English and 83.3% in French (by taking 1,000 words as test cases for the respective cases). The method is also tested for Spanish (with 124,726 words corpus), Latin (125,000 words corpus), Italian (100, 000 & 1,000,000 words) and reported success (Goldsmith, 2001a).

Viewed from high-rate precision reported, unsupervised method of morphological analysis seems competent. Moreover, Goldsmith (2001a) shows unsupervised systems can be made ***language independent***. This language independent nature of unsupervised learning is considered by Goldsmith (2001a) as a justification for the existence of 'universal grammar of

languages'[16]. Moreover, the approach pursued by Goldsmith doesn't require development of dictionaries and identification of rules of the language to develop a morphological system. Keeping all other factors constant, this seems to decrease greatly the task required in the development of morphological systems. For these reasons UL approach is considered for the current study.

2.3 Unsupervised Learning for Morphological Analysis

This section focuses on description of algorithms employed by Goldsmith (2001a) to develop Linguistica2001; which generates morphological dictionaries from a given corpus using an unsupervised learning approach. It performs the task by splitting words of a given corpus into word breaks (i.e., prefixes, suffixes, and stems) and put them in separate dictionaries. Then a connector, called signature, is created to relate the corresponding entries of the dictionaries. The system has also components to optimize the quality of word breaks identified. This optimization is made by applying some heuristics and conducting a Minimum Description Length (MDL) test. Because of practical importance MDL has in this regard, it is presented briefly in the following section.

2.3.1 Minimum Description Length (MDL) Framework [17]

MDL is a theory that is based on traditional wisdom that says *improved compression of the learning data samples leads to better generalization properties and better prediction on unseen data*. MDL has a theoretical background from *Kolmogrov Complexity*, which is proposed to identify the length of the shortest effective binary description of strings. Mathematically, MDL can be derived from Bayes's rule; but with additional consideration of

[16] It is an assumption in linguistics (esp. in generative grammar) that claims the existence of underlying principles (grammar) common for all languages. It is assumed that languages vary only in their surface forms.
[17] Detailed description is available in Vit'anyi and Li (1999).

the notion of *Universal Distribution* and *Martin-Löf test for randomness of individual objects (*Vit'anyi and Li, 1997, 1999).

MDL has the following assumptions (Vit'anyi and Li, 1999). If a body of data D does not contain any regularities at all, then it consists of purely random data and there is no hypothesis to identify. But assume that the body D contains regularities. With help of description of those regularities (a model or hypothesis), we can describe the data *compactly.*

Assuming that the regularities can be represented in an effective manner, we can encode the data as a program for that representation. Vit'anyi and Li (1997) pointed that squeezing all effective regularities out of the data; we end up with a representation intended to the meaningful regular information in the data together with a program for representation intended for the remaining meaningless randomness of the data.

With these consumptions, MDL is used in hypothesis selection (Vit'anyi and Li, 1997). Given a sample of data and an effective enumeration of models, ideal MDL selects the model *that minimizes the sum of*:

a) *the length, in bits, of an effective description of the model*
b) *the length, in bits, of an effective description of the data when encoded with the help of the model.*

The MDL theory states that with a more complex description of the hypothesis H, it may fit the data better and therefore decreases the misclassified data. If H decreases all the data, then it does not allow for measuring errors. A simpler description of H may be penalized by increasing the number of misclassified data. If H is a trivial hypothesis that contains nothing, then all data that described literally and there is no generalization.

However, the MDL test is used only for selection purpose. A system that applies an MDL test needs a component to generate different hypotheses of which MDL selects one. In the following section algorithms used by Goldsmith to generate morphological hypotheses (morphological splits) are described. The section also discusses how the MDL test is applied in the area of morphological analysis.

2.3.2 Unsupervised Learning Algorithms

Goldsmith (2001a) has used a series of heuristics to develop his unsupervised morphological analysis system: -

a) ***Initial Splitting*** – to identify a set of candidate word breaks - an optimal division of each word into stems and affixes[18].
b) ***Forming Signatures*** – identifying list of suffixes that appear with each optimal stem identified.
c) ***Optimization with Heuristics and MDL***– to identify regular morphemes that are acceptable some how for their linguistic quality
d) ***Triage*** – to further fine-tune the morphemic components identified.
e) ***Paradigm (word categorization)*** – to determine inflectional categories based on pattern of suffixation.

2.3.2.1 Initial Splitting

The initial splitting is done using a ***modified version of the Zelling Harris's algorithm of Successor Frequencies*** (Harris, 1955 as described in Goldsmith, 2001a). Harris algorithm helps to detect morpheme boundaries (given a phonemic representation) by asking, in between each letter of a word, how many different ways there were to finish a word, given all of the letters (or phonemes) up to that point in the word.

[18] In Goldsmith (2001a) initial experiment, only suffixes and stems are considered (see the formulas). But all that are meant for suffixes also works for prefixes. The split of words into stem and suffix is also arbitrary, i.e., do not convey their linguistic notations, and used to mean just the first and second parts of the split.

But Goldsmith identifies that this algorithm fails for a number of reasons. The algorithm, for example, is susceptible for peripheral word breaking due to the existence of large number of words beginning with the same sequence of letters. Goldsmith made a modification by giving more weight to word breaks that are around the middle or end of words than those at the beginning (Goldsmith, 2001a). This is done by including length information into the calculation of word break points. The task involves two processes: - computing the word break point values (referred to as '**figure of merit**' in Goldsmith (2001a)); and distributing the corpus frequency of each words to their respective word breaks.

To compute the figure of merit, Harris's Algorithm is applied first to extract all possible word breaks of the corpus words. Then for each possible breaks of a word W into $Stem_i$ (part of a word to the left of morpheme break) and $Suffix_j$ (part of a word to the right of a morpheme break), the figure of merit is calculated using the following equation[19]

$$H(stem/suffix) = -\left(|stem| * \log\ freq(stem) \quad + \quad |suffix| * \log(suffix)\right) \ldots\ldots\ldots\ldots\ (1)$$

Or more generally,

$$H(stem/suffix) = -\sum |morpheme_i| * \log\ freq(morpheme_i) \ldots\ldots\ldots\ldots\ (2)$$

Equation above uses length information such as length of stem '|stem|' and frequency information such as log frequency of a particular suffix in the corpus. For example, if two parses /go-vernments/ and /govern-ments/ are given to a corpus word 'governments' the figure of merit of each word breaks can be calculated in the following way. Assuming that {go} appears 200 times, {vernments} 50 times, {govern} 30 times and {ments} 500 times in the corpus, the figure of merit of the word break /go-vernments/ is (-

[19] In the equation more weight is given to word breaks in the middle or end position of a word by using the sum of product of the inverse log frequency and the length, in letter, of the word break.

(2*LOG(10/200)+(9*LOG(10/50))))[20]= 8.89; and that of /govern-ments/ is (-(6*LOG(10/30)+5*LOG(10/500)))= 11.36. With this calculation, the word break /govern-ments/ has better figure of merit (11.36) which shows that the heuristics helps to give more weight to word breaks in the middle or end of a word that those at the beginning.

The second process is distribution of the word's corpus frequency among its various word breaks. This is to mean if 'governments' appears 50 times, distribute this frequency value to all its word parses. Goldsmith uses ***Boltzmann distribution***[21] for this purpose, which is used to make the probability mass associated with a particular stem/suffix split S_i proportional to $e^{H(S_i)}$. Thus for any given word *w*, a normalization term Z is established, which is the sum over all parses P_i of the figure of merit, i.e., $Z = \sum_i e^{H(S_i)}$. Then the model assigns a probability, to any given split of a word, of $\frac{1}{Z} e^{H(S_i)}$.

However, a word with complex morphological structure might not be analyzed completely by one run. To parse such complex words as 'bureaucratization' - /bureau-cracy-ize-ation/ the above two processes are required to run iteratively. In Goldsmith's experimental case, less that 5 iteration were enough to analyze all words in the corpus (Goldsmith (2001a).

2.3.2.2 Identification of Signatures

Once the initial splitting is complete, word splits are put in separate lists, and a signature is created to link stems with the associated suffixes or prefixes. Signatures are used to organize the stems and suffixes in such a way that the words in the corpus can be generated from the

[20] The word 'managements' is 11 characters long. Assuming each position between two consecutive letters of the word as potential word break points, the word has 10 word break points.
[21] The Bolzmann distribution describes the relative probabilities of finding a system in different states as a function of temperature.

broken components. For example the English words *despair, pity, appeal,* and *insult,* appear with suffix *ing* and *ingly.* They are also used with no suffixes (i.e., with NULL suffix). Thus they have a common signature ***NULL.ing.ingly***.

However, such a signature is created only for stems and suffixes that are optimal at least to one word in the language. This is to mean that /-ed/ is kept in the signature as suffix only if it appears at least with one word say with /work/. This is done to eliminate spurious parses. Such word breaks are identified automatically by computing ***log (stem count) * log (affix count)*** and taking only those word breaks with non-zero values.

Goldsmith (2001a) referred to the remaining signatures as ***regular signatures*** and the associated suffixes as ***regular affixes*** assuming them as possible morphemes. *Though* such regular affixes constitute a good initial analysis, Goldsmith states that they are not quite the affixes that we would like to establish for a natural language. In other words, to make them linguistically acceptable, further optimization is required. Some heuristics he used for this purpose and the associated MDL test designed to ascertain the improvement is presented in the following section.

2.3.2.3 Optimizing the Morphology using heuristics and MDL

By application of initial splitting and signature creation algorithms, words of the corpus are segmented into different pieces (stem, suffixes), and a signature is formed to maintain the association that exist between them. In doing so for the entire words of the corpus, lists of stems, suffixes, and signatures are established. These lists can be considered as the morphology of a given language - the language of the corpus (Goldsmith, 2001a).

With such morphological system, ***a word having a stem and suffix components can be identified by a set of pointers*** – i.e., by a pointer to a signature and pointers within the signature to the stem and suffix components. For example, if {work} and {-ed} are available in the stem and suffix lists, and **NULL.ed.ing.s** is a signature that associates the two morphemes, the word /worked/ can be identified by a pointer to the signature **NULL.ed.ing.s** (which is stored in the signature list) and by the two pointers within the signature that point to {work} of the stem list and {-ed} of the suffix list.

Goldsmith (2001a) applied the MDL test based on such conception of morphological representations. As stated in 2.3.1, the MDL framework selects a hypothesis (or a model) that minimizes the sum of the compressed length of the description of the model and the compressed length of the description of the data represented using the model. In the case of morphological system, the different lists produced (such as stem, suffix, and signature lists) can be considered as the model; and the pointer-based word representations as the data representation applying the model (Goldsmith, 2001a).

Hence, the **compressed length of the model** (or **of the morphology** in our specific case) can be computed from the sum of the lengths of the stems, the suffixes, the signatures and some organizational overhead[22]. But computer ***representation of lists demands two things – the contents that form the list, and the organizational structure used to keep the list intact***. Goldsmith uses **pointers** as the organizational structures of his lists. With this representation, the suffix list consist list of pointers to the suffixes of the language and the list itself of those suffixes. Similarly, the stem list holds list of pointers to the stems of the language and the list itself of those stems.

[22] represents information that normally would be built into the graphical structure of the model

However, the signature component is more complex. As the rest of the lists, it has list of pointers to the signatures of the language. But the lists of signatures themselves are lists of pointers pointing to stems and suffixes stored in the stem and suffix lists. Moreover, as some stems can be complex (they can have nested stems and suffixes), additional mechanism should be included to distinguish complex signatures from the simple ones. In addition, Goldsmith uses frequency information for a number of applications. Thus, each signature has also structures to hold counts of stems and suffixes appearing with the signatures. This makes the representation of signatures somewhat complex. Despite this, Goldsmith calculates lengths of the different lists as follows. In the calculation, he uses the following notation: -

W:	set of all words of a lexicon or of a corpus
W_{simple}[23] :	set of words with simple stems
$W_{complex:}$	Set of words with complex stems
f:	suffix
t:	stem
σ:	signature
[W]:	token-count, either of a specific word, morpheme, or a category depending on a context
<item>	type-counts, i.e., the number of orthographically distinct item

(1) Compressed length of morphology

(*i*) $\log < suffixes > + \log < stems > + \log < signatures >$ [24]

(*ii*) *Suffix list* $\sum_{f \in Suffixes} \left(\lambda^* \mid f \mid + \log \frac{[W_A]}{[f]} \right)$

(*iii*) *Stem list* : $\sum_{w \in W_{UN} \cup W_{SIMPLE}} \left(\lambda^* \mid stem(w_i) \mid \quad + \quad \log(\frac{[W]}{[stem(w_i)]}) \right)$

(iv) *Signature component*

[23] W_{SIMPLE} refers to stem with no nested morphological structure while W_{COMPLE} contains such structure
[24] n items can be represented using $\log_2$ (n) bits of information

Stated once for the whole component:

(a) Signature list:

$$\sum_{\sigma \in Signatures} \log \frac{[W]}{[\sigma]}$$

For *each* signature:

(b) Size of the count of the number of stems plus size of the count of the number of suffixes: log <stems (σ)> + log <suffixes (σ)>

(c) A pointer to each stem, consisting of a simple/complex flag, and a pointer to either a simple or complex stem:

(i) Case of simple stem: flag of length $\log \frac{[W]}{[W_{SIMPLE}]}$ plus a pointer to a stem of length $\log \frac{[W]}{[t]}$; or

(ii) Case of complex stem: flag of length

$\log \frac{[W]}{[W_{COMPLEX}]}$, followed by a sequence of two pointers of total length $\log \frac{[W]}{[stem(t)]} + \log \frac{[\sigma]}{[suffix(t)\ in\ \sigma]}$.

(d) a pointer to each suffix, of total length $\sum_{f \in Suffixes(\sigma)} \log \frac{[\sigma]}{[f\ \ in\ \sigma]}$.

(2) Compressed length of corpus

The compressed length of the corupus is calculated as:

$$\sum_{w \in W} [w]_{raw} \left[\log \frac{[W]}{[\sigma(w)]} + \log \frac{[\sigma(w)]}{[stem(w)]} + \log \frac{[\sigma(w)]}{[suffix(w) \cap \sigma(w)]} \right] \quad \text{.....(3)}$$

In the calculation, length of pointers is calculated based on the assumption[25] that the length of a representation of an object of raw frequency F selected from a well-defined set of total frequency T is equal to log T/F. With this understanding, the length of the pointer to a suffix *f*, for instance, is represented as $\log \frac{[W]}{[f]}$. Moreover, in the calculation of the length of stem and suffix, Goldsmith employs a factor of proportionality λ, which he takes to be log (A), where *A* is the number of letters in the alphabet employed in the corpus.

The formulas in *i, ii, iii* and *iv* are used to calculate different components required to calculate the length of the morphology[26]represented in (1). As discussed above, individual member of the signature list have pointers to a stem and suffix. The stem part also consist a simple/complex flag, and a pointer to either a simple or complex stem. In the case of complex stems, the flag is followed by three pointers: a pointer to a signature σ, a pointer to a stem *t* within the signature, and a pointer to a suffix *f* in the signature; these three pointers jointly specify the word which serves as the morphological base of the present word. These three pointers are, respectively, of length: $\log \frac{[W]}{[\sigma]} + \log \frac{[\sigma]}{[t]} + \log \frac{[\sigma]}{[f \ in \ \sigma]}$.

The compressed length of the corpus (or the data) is measured using the formula in (2). Since words in the corpus are represented using pointers to a signature and pointers within the signature to stems and suffixes, length of a word is computed from the length of these three pointers.

[25] taken from Information Theory
[26] Formula (*i*) is used to calculate the length of the organizational overhead.

Using these two equations, two separate measures – with and without including a signature σ as part of the morphology, are conducted for each signatures of a language. If the MDL value (the sum of the two lengths) minimizes with the test conducted by deleting σ, then σ is not considered as good signature and hence removed permanently. This is because inclusion of σ as a signature increases the sum of the lengths of the morphology and the corpus representation; which in the MDL principle reduces the predicting power of the hypothesis (Vit'anyi and Li, 1999).

Goldsmith reported that the MDL test helps to remove a number of spurious results. However, he also found in his experiment that some of the outputs found after the MDL test are not acceptable by human experts. To deal with such outputs, he included a triage algorithm in the system. Triage algorithms are discussed in the following sections.

2.3.2.4 Triage

The goal of *triage* is to determine how many stems must occur in order for the data to be strong enough to support the existence of a linguistically real signature. Triage is done by removing certain signatures and observing the improvement gained. In Goldsmith's experimental case, any signature for which the total number of stem letters is less than 5, and any signature consisting of a single, one-letter suffix are deleted; he keeps, then, only signatures for which the savings in letter counts is greater than 15 (where *savings in letter counts* is simply the difference between the sum of the length of words spelled out as a monomorphemic word and the sum of the lengths of the stems and the suffixes); 15 is chosen empirically

2.3.2.5 Determining Paradigms[27]

The forgoing discussions focus on methods of finding linguistically acceptable morpheme lists from a given corpus. The next step is to identify word paradigms (inflectional categories) based on the existence of a regular pattern of suffixation with *n* distinct suffixes. For example, with a regular paradigm in English consisting of the suffixes *NULL, -s, -ing,* and *-ed*, we expect to find verbal stems while with suffixes like NULL, -s, -ist, we expect to get nouns like man, cows, an scientist. This type of word paradigm determination is also linguistically acceptable (Williams, 1981).

2.4 Remaining Issues for Semitic Morphology

Section 2.4.2 describes the algorithms used in the development of Linguistica2001. This system has been tested with English, French, German, Spanish, Italian, and Latin corpuses and reported functional (Goldsmith, 2001a). However, the initial splitting algorithm is designed to parse morphemic components having linearly arranged phonemic components. As a result, it is difficult to apply the system for analysis of the non-sequential morphemic components of Semitic stems.

Thus, application of the Goldsmith's unsupervised method of morphological analysis demands inclusion of a component to handle morphological processes internal to Semitic stems. For this purpose, a method devised in ***Autosegmental Phonology*** is used. The method is used by a number of researchers in representation of Semitic morphology (McCarthy (1981), Beesley (2000), Kiraz (1995 & 1994)). This method is discussed in more detail in the following section.

[27]paradigm refers to all inflectional variation of a base work like *love, loved, loves, lovely* and *loving* of the base word 'love'

2.4.1 Autosegmental Representation of Semitic Morphology

In autosegmental approach, different phonological elements (also called autosegments) such as tone, skeletal (CV-), and stress are represented in different tiers. Then relation between elements of different tiers is represented by association lines (Crystal, 1991). The following is an example.

Figure 2-1: **Example of autosegmental representation in phonology**

In the example there are three tiers – tonal, skeletal and segmental. The segmental tier represents the actual sound segments, while the skeletal tier shows the CV-pattern. In the above diagram, the solid line drawn from [m] of the segmental tier to C of the skeletal tier shows a relation between them, that is, it shows that [m] is a consonant.

Such a representation system is also adopted in the representation of Semitic morphology. McCarthy (1981) for example used it to represent Arabic morphology. He defined a type of morpheme found in Semitic stem as an ordered string of 1 x n feature metrics associated autosegmentally with a root node μ. McCarthy (1981) shows that the root node μ identifies this string as a particular morpheme. In addition, μ bears all morphological information associated with the morpheme, such as rule diacritics, whether it is a root or an affix. Other

features of the morpheme hold separate tiers. With such representation, the Amharic stem /säbbärä/ ‘he broke’, for instance, can be represented in the following way.

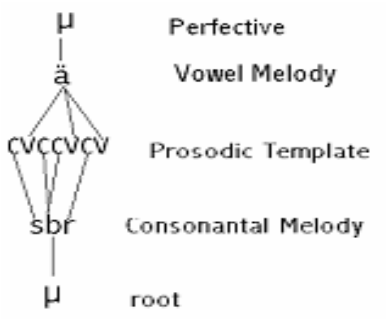


Figure 2-2: An example of Autosegmental Representation

The figure shows that the perfective /säbbärä/ is produced from a root {sbr} ‘break’ and a vocalic pattern {ä} following a CVCCVCV stem template. Each of these features is represented in separate tiers and the relation that exists between them is represented by association lines.

In this way, stem components identified by the MDL-based system can be further analyzed into its subcomponents.

CHAPTER III

THE STRUCTURES OF AMHARIC WORD

This chapter discusses Amharic word morphology giving more emphasis on the description of morphological processes involved in word formation. It also presents brief account on Amharic phonetics and word classes. Most of the idea and examples presented are taken form Baye (1999, 1994), Getahun (1997), Girmay (1992), Podesky (1984, 1986), Mullen (1986), and Habte Mariam (1994).

3.1 The Amharic Phonetics

Amharic has 33 consonants and seven vowels. Table 3.1 shows these consonants along with their place of articulation and phonemic features (adopted from Mullen (1986).

		Labials	Dentals	Palatals	Velars	Labio-Velar	Glottals
Stops	Voiceless	p ፕ	t ት	č ች	k ክ	k^w ኳ	◁
	Voiced	b ብ	d ድ	ğ ግ	g ግ	g^w ጓ,	
	Glottalized	P ጵ	t' ጥ	č' ጭ	k' ቅ	k'^w ቋ	
Fricatives	Voiceless	f ፍ	s ስ	š ሽ			h
	Voiced		z ዝ	ž ዥ			
	Glottalized		s' ጽ				
	Rounded						h^w ኋ
Nasals	Voiceless						
	Voiced	m ም	n ን	ñ ኝ			
Liquids	Voiceless						
	Voiced		l ል, r ር				
Glids		w ው		y ይ			

Table 3-1 Amharic Consonants

The seven vowels of the language are 'i, e, a, o, u, ä and ◁'. Depending on their point of articulation, Amharic vowels can be categorized into peripheral and central vowels. The five peripherals are 'i, e, a, o, and u – the first two are back vowels (articulated at the back of our

mouth) while the other are referred as front vowels. The predominantly used vowels of the language are however the central □ and ä.

There are three major phonological processes commonly observed in Amharic consonants. These are gemination, palatalization and labialization[28] (Bender and Hailu, 1978). Consonant gemination in Amharic is both lexical and morphological. Lexical gemination is observed in such words as /gäna/ 'still' and /gänna/ 'Christmas'. The difference in the meanings of the two words comes only because of the geminated [n] occurred in the second word. Morphological gemination occurs in conjugation of verbs like in the perfective stems such as /säbbärä/ 'broke' and /wässädä/ 'took'. Labialization (rounding) affects every consonants preceding [o] as in /q^{w}ommä/ 'stop' and /g^{w}ottätä/ 'pull'. However, palatalization is restricted to dentals in deverbalization processes (Baye, 1994). Examples are presented below.

/wäsd-/'take' /wäsäd-i/ ➔ [wäsäj] 'taker'
/tärrt-/ 'narrate' /tärrät-i/ ➔ [tärräč] 'narrator'

The example above shows when dentals such as [d] and [t] are followed by the back vowels like [i] and [e], they are changed to their corresponding palatals [j] and [č] due to phonological factors.

In Amharic, clusters of two consonants are allowed around the middle and end pats of words. Initial clusters are highly restricted. In the case of impermissible consonant clusters, the vowel of epenthesis [□] is inserted (Mullen, 1986).

[28] Gemination refers to a sequence of identical adjacent segments of a sound in a single morpheme whereas palatalization is a secondary articulation involving a movement of the tongue towards the hard palate. Labialization is a secondary articulation involving any noticeable lip-rounding as in the initial [k] of 'coop', which is labialized because of the influence of the labialization in the following vowel [u] (Crystal, 1991).

3.2 Amharic Word Classes

There are two common methods to identify lexical classes of a word (Williams, 1981). The first is based on the type of suffixes a particular word takes. For example in English nouns can not take {-ed} as their suffix but verbs do take. The second method is based on the position of a word in a sentence. For example, in Amharic a verb cannot come at sentence initials and a noun cannot come at the end of sentences (Baye, 1994). Based on these two criteria, Amharic words can be categorized into the following five word classes: -

a) **Noun 's◌́m'** – comprises words that take {*-occ̆}* as a plural marker. This group also includes words like /b◌́rhan/ 'light', and pronouns like /◌́nne/ 'I', /anta/ 'you', and /ïrs⁺a/ 'she'.

b) **Verb 'g◌́s'** – Verbs are those words that come at sentence ends. They also take Clitics like the 2sg.m. {-h}, 1sg.c. {-hu}, 3sg.f. {-č} and the like.

c) **Adjectives 'k'◌́s's'◌́l**' – are words that come before nouns in sentences. Adjectives serve to modify nouns they precede. In a sentence '◌́su bät'am gobäz l◌́j näw' – 'he is a very clever boy' there are two adjectives /bät'am/ and /gobäz/. 'bät'äm' 'very' modifies /gobäz/ 'clever', and /gobäz/ modifies the noun /l◌́j/ 'boy'.

d) **Adverb 'täwsakä-g◌́s'** – words in this group are small in number and they do not attach any kind of prefixes and suffixes. The only words belonging to adverbs are gimäňa, j◌́läňa, m◌́näňa, k◌́fuňa, ◌́ndägäna, and gäna (Getahun, 1997).

e) **Prepositions 'mästäwadäd'** – comprises words that are used to form adverbial phrases appearing before nouns they serve. Prepositions could not serve as base for generation of other words nor do they conjugate for any kind of grammatical formation

as for number, gender, etc. {lä}, {kä}, {bä}, {yä}, {slä} are some prepositions of the language.

3.3 Word Formation

Amharic words are believed to have a mono-, bi-, tri-, quadri-, quinqui- or sexi-radicals consonantal roots[29] that carry their semantic information (Bender and Hailu, 1978). The following are examples: -

Mono-radical:	ša	'desire'
bi- radical:	täw-	'leave'
tri- radical:	säbr-	'break'
Quadri - radical:	gälb□t	'turn over'
Quinqui -radicals:	-šḱänäťť-r-	'throw away violently'

Different word forms are formed from such consonantal roots. Mullen (1986) has indicated that word formation in Amharic involves three processes: - ***creation of verbal stems from consonantal roots, production of fully inflected word forms from stems and cliticization.*** Amharic verbal stems are formed by infixing vocalic melodies into consonantal roots. Then the verbal stem formed undertakes inflectional process for a number of grammatical contexts such as tense, number, gender and the like (Girmay, 1992). Fully inflected verbal stems can then attach Clitics. See the following examples:-

Consonantal root:	{sbr} 'break'
Stem formation:	{sbr} + {ä} ➔ /säbbär-/ 'broke'
Inflection:	/säbbär-/ + 2sg.mS. ➔ /säbbärk/ 'you broke'
Cliticization:	/säbbärk/ + 1sg.cO. ➔ /säbbärkäňň/ 'you broke me.'

The example above shows the three levels involved in word formation. In the first level, a perfective stem /säbbär/ is formed by inculcating the vowel {ä} into the root {sbr}. Then /säbbär/ is inflected for the second person singular masculine (2sg.mS) by adding the suffix {-

[29] Baye (1999) however argues that Amharic has only three radical roots.

k} to form the inflected stem /säbbärk/. Then the first person object maker (1sg.cO) {äňň} is cliticized to form /säbbärkäňň/ which adds information that the speaker is the object of the breaking action and the listener is the action source.

However, the example given is one of the many forms of word formation used in the language. Moreover, there are phonological processes involved in each steps of the word formation that make a change in the word form produced (Mullen, 1986). The following sections give further accounts on the subject.

3.3.1 Stem formation

Baye (1999) states that there are two type of morphological processes that occur in stem formation – ***radical reduction*** and ***radical extension***. Radical reduction is a process of losing one or more radicals during stem formation. Amharic has some set of consonants referred to as ***weak radical*** which are missed during stem formation. These set of radicals are /h, y, w, ◁/, and marginally, [r] and [b] (Baye, 1999). The following examples show verbal stems involving radical reductions: -

Root	**verbal**	**deverbals**	**gloss**
f-r-h	färra	f□rh-at	'fear'
s-y-ť	šäťä	š□yyaĉ	'sale'
q-w-m	q⌖omä	q□wwame	'opposition'
l-◁-k	lakä	l□◁uk	'delegation'

An ***extension*** is a derivational process that involves insertion of new radicals and/or extension of existing radicals. There are both ***internal*** and ***external extensional processes*** (Baye, 1999). Internal extension is an extension of any one or two of the radicals of a root. Of the two, internal extension is discussed more in a number of literatures (Baye, (1999), Leslau (1995),

for example). The two common examples of internal extension are ***gemination*** and ***redistribution***. They are discussed in more detail in the following sections.

3.3.1.1 Gemination

Amharic roots geminates for a number of reasons. Baye (1999) shows that in three radical Amharic roots, any one of the three radicals[30] undertakes gemination process. Radical gemination is observed in grammatical features such as attenuatives, intensive, and adjutative stems of adverbs. Amharic verbs also show gemination in perfect and imperfect (Wedekind, 1994 and Habte Mariam, 1994). Each of these are discussed below: -

Attenuatives and intensive stems

The intensives and attenuatives stems are used as adverbial quantification. These stems are derived with the gemination of the ultimate and the penult radicals to express such adverbial notations like degree of intensity or iteratively of actions. For example, in sentence 1 below, the gemination of the penult and ultimate radicals of /k□ff□tt-/ shows the violate opening of the door whereas in 2 / käfätt-/ shows that the door is opened gently.

(1.) Kasa bärr- u- n k□ff□tt adäräg – ä – w
K. door – def – acc. open violently do – pf - 3mss -3msO
'Kasa opened the door violently'
(2.) Kasa bärr- u- n käfätt adäräg – ä – w
K. door – def – acc. open slowly do – pf - 3mss -3msO
'Kasa opened the door violently'

[30] For a three-radical root 123, radical 1 is known as initial, 2 is a penult and 3 is an ultimate radical.

Addaragi (adjutative)

The initial radical of Amharic stems could also geminate in adjutative to express participation of a subject in a particular action. In the example[31] below, /a-ggadäl-ä-w/ shows an adjuative feature. It shows that Kassa has participated in the killing of the lion.

Kasa Ayyälä-n anbäsa a-ggadäl-ä-w
K. A. – acc lion adju kill-pf-3mss-3mso
'Kassa helped Ayele kill a lion'

Reciprocal

This is a grammatical feature that shows that two or more subjects take action on one another. In the example below, /tä-gäddäl-u/ indicates the killing of one another.

Kasa □nna Ayyällä tä-gaddäl- u
Kasa and Ayele rcp- kill- pf 3pl
'Kasa and Ayele killed each other'

Reciprocal causative

Baye (1999) shows that reciprocal causative (addaragi) derived form /a-ggaddäl-/ is derived from the concatenation of /a-t/gaddäl-/ in which the reciprocal morpheme /t-/ assimilates to the initial radical /g/ of the stem to form the form [aggadäl-] 'caused kill one another'.

Aster Kasa -n □nna Ayyälä-n a- t- gaddäl –äčč-aččäw [aggaddäläččaččäw]
Aster Kasa acc. & Ayyala acc. cs-rcp-kill 3fs- 3pl
'Aster caused/made K. and A. kill one another'

In the example the reciprocal /-t-/ is assimilated. /a-/ is a causative prefix which shows that Aster is the cause for the one another killing of Kassa and Ayyälä.

[31] The examples are taken form Baye (1999)

Manner nominal stems

Manner nominal stems show the manner in which something is done. Such a stem is shown by gemination of the initial radical and reduplication of the penult radical of the root, the later showing the iterative nature of the action or event. Consider the following examples:

Roots	**perfective stems**	**nominal stems**	**manner forms**	**gloss**
l-b-s	läbabbäs-	-lläbabäs	a-lläbabäs	'manner of dressing up'
m-ť-◌	mäťať-	-mmäťať	a-mmäťať	'manner of coming'
b-l-◌	bälal-	-bbälal	a-bbälal	'manner of eating'

In the examples above, /alläbabäs/, /ammät'at'ä/ and /abbälal/ are nominal stems. The manner nominal stems have a geminated first radical and a redistributed second radical.

Gemination in stems with dropped radicals

Typical features of gemination are also observed in stems that drop one or more of their radicals. Consider the following examples: -

Roots	**perfective stems**	**nominal stems**	**adjutative stems**	**gloss**
h-r-s	härräs	[arras]	as-t-ärarräs-	'manner of farming'
h-r-m-	härräm	[arräm]	as-t-ärärräm-	'manner of weeping'
◌-č-d-	◌äččäd-	[aččäd-]	as-t-äčaččäd-	'manner of harvesting'
◌-k'-f-	◌äk'k'äf-	[ak'k'äf-]	as-t-äk'äk'k'äf-	'manner of hanging up'

As shown in the list above, all stems lost one of their radicals. Such stems add the prefix {as– } to form the adjutatives.

3.3.1.2 Reduplication

Another morphological operation involved in stem formation is reduplication. There are two types of reduplication - ***total*** and ***partial*** (Baye, 1999).

Total reduplication

Total reduplication shows iterative actions of attentuatives and intensive manner. The two type of iterative (attenuatives and iterative) are formed by total reduplication of the corresponding attenuative and intensive stems. Such stems combine with the auxiliary verb /al-/ 'say' or /adarrag-/ 'made' to form compound predicate of the type shown in the following examples: -

Kasa mästawät- u- n s□bb□r-s□bb□r adäräg – ä – w
K. mirror – def – acc. Break-break do – pf - 3mss -3msg.O
'Kassa broke the mirror into pieces'
mästawot -u s□birr-s□bb□rr al-ä
glass -def break-break say-pf.3m.sg.S.
'*The glass broke into pieces'*

The reduplication s□bb□r-s□bb□r in the two sentences above indicates the intensity of the action – the mirror is broken into many pieces. The attenuative and intensive stems themselves are derivatives, which has undergone gemination of the ultimate, or penult or both radicals.

Partial reduplication

Unlike total reduplication, partial reduplication does not duplicate the entire radicals. It rather duplicates one or two of the radicals of the root. Baye (1999) presents three degree of partial duplication: - roots duplicate only its penult to show iterative actions; roots duplicate both the ultimate and the penult radicals, and the penult is geminated, and the third degree, an extension of the second in which radicals continue duplicating themselves indefinitely.

Example:

Roots	**attenuatives**	**intensives**	**gloss**
1. s-b-r	s-b-b-r	säbabbärä	'break into pieces completely'
2. k'-m-s	k'-m-m-s	k'ämammäsä	'taste repeatedly/lightly
3. s-b-r-	s-b-r-b-r	s□b□rb□rr	'break into pieces completely
4. f-r-s-	f-r-s-r-s	f□r□sr□ss	'demolish completely'
5. s-b-r-	s-b-r-b-r-b-r...	s□b□r□b□rr…	'break into pieces and pieces
6. f-r-s-	f-r-s-r-s-r-s…	f□rs□r□sr□ss…	'demolish into rubbles'

In example 1 and 2, the second radical (penult) is distributed to show iterative actions. In example 3 and 4, however, both the penult and ultimate (last radicals) are duplicated and show the intensity of an action. In examples 5 and 6, the speaker of the sentence can extend the reduplication as long as he feels enough to show the way an action is performed. The example blow, for instance, is used to show the way Kasa broke the glass.

Kasa b□rĉ□k'o-w – n s□b□rb□rb□rr… adärräg – ä – w
K. glass –def – acc break, break, break do- pf- 3m.sg.S –3m.sg.O.
'Kasa broke the glass into pieces and pieces, and pieces ….'

The example above shows that the glass is broken into a number of pieces. In these ways, reduplication is used in stem derivation – both from root, and from stems which themselves undertake derivation. In the following section we see basic templates used to form Amharic stems.

3.3.2 Some points on Amharic Stem Templates

Amharic has complex morphological structures (Mullen, 1986 and Girmay, 1992). Its complexity comes from the number of morphological and phonological processes involved in word formation. However, there is a pattern which makes the word formation process manageable to understand. This pattern is the templates following which stems are formed. As it is stated in Mullen (1986) Amharic has six verbal stems – perfective, imperfect, gerundive, jussive, infinitive, and agentive. These verbal stems provide the basis for all tense, aspect, and modals of the verb. The verbal templates are listed below taking {sbr} 'break' as exemplary.

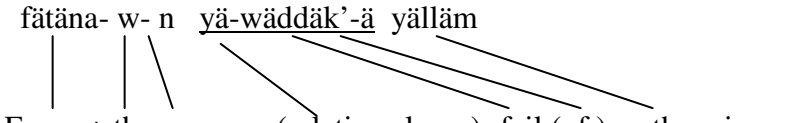
Stem	Template	Example
Perfective	CVCCVC-	säbbär-
Imperfective	-CVCC-	-säbr-
Gerundive	CVCC-	säbr-
Jussive	-CCVC	-sbär
Infinitive	-CCVC	-sbär
Agentive	CVCVVC-	säbaar-

However, variations in the structure of the templates could occur depending on the type of verbs[32] that uses the template. Despite this, all verbal stems are formed using these templates. A stem formed in this way can also undertake inflectional processes. The following section gives more on this.

3.3.3 Amharic Inflections (Affixation)

Abiyot (2001) has discussed Amharic prefixes and suffixes in greater detail. This section gives summarized account referring readers to chapter 3 of Abiyot (2001) for detailed presentation.

Amharic has a number of prefixes and suffixes that attach onto stems in the normal concatenative manner. Two or more prefixes/ suffixes could also come together. Consider the following sentence (taken from Abiyot, 2000).



(There is no one who failed the exam)

From the sentence /-wn/ of /fätänäwn/ and {-ä} of /yäwädäqä-ä/ are suffixes while {yä-} takes the position of a prefix. /-wn/ in the first word is a concatenated suffix. It is formed from {-w}, a definite, and {-n}, an accusative.

[32] verbs are categorized into Type A (geminates there penult only in the perfect) Type B(geminate their penult in all the cases) and Type C (geminate their penult only in the imperfect and perfect) see Bender and Hailu (1978)

Based on their functions, Amharic verbal prefixes can be classified in to clause markers, negative marker, subject markers. There are also a number of suffixes that can be used for a number of grammatical functions such as gender, number, person, and the like.

In summary, this chapter presented morphological property of Amharic words. Amharic words are considered to have consonantal roots of various lengths (1 to 6 radicals). These roots are then used to form stems by infixing vocalic melodies. A number of morphological processes are also involved in stem formation. These processes can be summarized into radical reduction and extensions. Gemination and redistributions are the most commonly observable extension processes whereas some set of Amharic consonants (referred to as weak radical) are known to get lost during stem formation. Moreover, stems could inflict for a number of grammatical cases and/or take a number of Clitics. Finally, the six common verbal stems of Amharic discussed are perfective, imperfective, gerundive, jussive, infinitive and agentive.

CHAPTER IV

Corpus preparation and Algorithm Design

In this chapter, the corpuses prepared and the stem-internal morphological analyzer designed are discussed. The chapter is organized in three sections. Section one discusses the corpus preparation, problems encountered and the corrective action taken to tackle them. The next section describes the algorithms developed to handle stem-internal morphological processes. The last section discusses data structures designed to serve as tiers for the autosegmental representation of Amharic stem.

4.1 Corpus Preparation

In this research two corpuses are used. The first is used for the analysis work as required by Linguistica2001[33]. The system develops a morphological dictionary (referred to as **signature)** by analyzing words of a given corpus. Because Linguistica2001 could not produce linguistically correct stems in sufficient number for the **stem analyzer module** developed (for reasons stated in the next chapter), a small corpus comprising some 326 Amharic stems is also developed.

Lingusitica2001 needs a large corpus (in ASCII format) with a size ranging from 5,000 to 1,000,000 words (Goldsmith, 2001a). The corpus can be any natural document such as those found in newspaper and books. However, such a large Amharic corpus is not readily available. The required corpus is thus prepared from scratch in this study. The size, however, is very small compared to the corpus size used in other experiments. This is mainly due to the absence

[33] Linguistica2001 is downloadable from
http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/linguistica2001.exe

of ASCII equivalent for Amharic letters that necessitated transcription of the corpus in Latin or other alphabets having parallel ASCII codes. In this connection, an effort is made to use the standardized phonetic symbols of the language (see Table 3.1). Where the symbol has no ASCII equivalent, it is represented by Latin characters that have close similarity with the phone in question (Table 4.1 lists some changes made during transcription)[34].

Amharic Letter	Phonetic Representation	Text (ASCII) Conversion	Problem Encountered	Symbol changed to
ቅ	k’	k'	' is considered as a word separater	q
ኝ	ň	n	Already used	N
ች	č	c		c
አ	◌	(	‘a’ is use both as a vowel and consonant	a
ጅ	⌘	?	Semantic difficulty to relate ‘?’ with ‘ጅ’	j
ጥ	t’	t'	' is considered as a word separator	T
ፍ	č'	c'	' is considered as a word separater	C
ጵ	p’	P'	' is considered as a word separater	P
ጽ	s'	s'	' is considered as a word separater	S
ቭ			Not found in Amharic Consonants	v
Labialized Consonant (e.g. ኗ)	nʷ	nw	Represents two consonants	u
አ	□	o	Already used	ï

Table 4-1: Changes made in phonemic representation of Amharic alphabets

Column 1 of Table 4.1 shows the Amharic phones (alphabets). The phonetic representation of these alphabets is presented in column 2. However, as observed in column 3 these letters have no equivalent ASCII codes. To correct these problems, the symbols in the last column are used in place of wrongly represented phonetic transcriptions.

There is also another problem encountered. For example, a superscripted w (as in hʷ) is used to represent a labialized [h]. This is to mean that hʷ should be considered as a single phoneme (Podolsky, 1984). However, when it is converted automatically into ASCII format it is

[34] See Appendix 1 for complete list of character transcription used in this study

considered as two consecutive consonants – [h] and [w].To retain the labialized sense in the representation; [u] is used in place of w. This is done because C◆C[35] can be represented using CuC in Amharic without much variation in meaning (Podolsky, 1984). As an example – both /k'◆omä/ and /k'uomä/ give the same meaning, i.e., 'he stood up' or 'something terminated'.

Another related problem is the case of the glottal voiceless consonant [ʔ]. In the Amharic literature reviewed such as Podolsky (1994, 1996), Baye (1999) and Girmay (1982), one can see that its place has been taken by vowels as in /ʔnnat/ 'mother', /abbat/ 'father' and /antä/ 'you'. Following the same pattern, the respective vowels are used in place of this consonant.

With these corrective actions, a corpus of 5,236 words (a word by word count) is prepared for the experiment using Linguistica2001. The corpus is prepared by transcribing the first 50 pages of "Kä-admas bašagar" a fiction by Be'alu Girma. This fiction is known for its literal quantity. From this, it is assumed that the work possesses all word classes of the language in sufficient number. A manual check done on the 1st page of the book also confirms this fact[36]. The transcription is made by a 4th-year student with assistance from a postgraduate student of Linguistics. The correctness of the transcription is checked by a linguist[37].

The second corpus is prepared for the stem-internal morphological analysis task. The corpus comprises 326 stems that are collected from a number of articles and books on Amharic linguistics such as Lesleau (1995), Baye (1999), Girmay (1982) and Hundson (1999). The stems are collected by the researcher himself. During the collection, an effort is made to

[35] C stands for consonant
[36] From 87 words found on the page, there are 37 nouns, 16 adjectives, 6 adverbs, 25 verbs and 25 prepositions. Since prepositions are mostly found attached on words, they are not included in the first count.
[37] The correctness of the transcripition is checked by Dr. Zelealem Leyew and Ato Daneil Abera, Depratment of Linguistics, Addis Ababa University

consider the composition of productive words (words that are used as a base for other word formation) of the language. Since verbs are the most productive word classes of the language (Bender and Hailu, 1978), the number of verbal stems is large in the corpus. The corpus comprises 149 simple verbs, 63 derived verbs, 38 nouns and 76 adjectives.

However, the number of activities involved in the corpus preparation took considerable time of the research (about a month). Moreover, the corpuses prepared, especially the one meant for Linguistica2001, are very small in size. These two factors have their own impact both on the morphological system developed and on the experimental results found. Due to time limitation encountered, the researcher is unable to integrate Linguistica2001 with the stem analyzer developed, as a result of which, the test is conducted in two separate systems. There are also some problems (errors) in the experimental results, particularly in the stem components produced by Linguistuca2001, occurred as a result of the small size corpus used in the study.

4.2 Algorithm Design

The algorithms discussed in this section are meant for the autosegmental representation of Amharic stems. They are designed to identify the morpheme components of stems, to represent morphemic components in different autosegmental tiers and to maintain the association between related morphemes in different tiers. The last task is referred here as construction of stem signatures adopting the terminology used in Linguistica2001.

The algorithm in Figure 4.1 shows the general algorithm for autosegmental representation of stems. It is a general algorithm in a sense that it shows the general procedure followed in the construction of an autosegmental representation of stems. It shows the steps starting from reading a corpus file to the final representation of morphemic components in different

autosegmental tiers and making links between them. It calls other modules to accomplish the tasks listed.

1. Open a corpus file,
2. read a stem
3. identify the morpheme components of the stem
4. add the identified morpheme components into respective tiers of the autosegmental representation
5. construct a stem signature
6. if end of file, job accomplished (end), else go to step 2

Figure 4-1: An algorithm for Autosegmental representation of Amharic Stems

Most programming languages have inbuilt module to handle file operating tasks (opening a file in a read/write/append mode, reading/writing from/to files and the like). Thus, the first task can be accomplished using these inbuilt modules. However, the rest of the activities in Figure 4.1 require writing functions to handle the respective tasks.

Reading a word, for example, demands to know what constitute a word and what delimits a word. In this regard, a word is considered as a sequence of characters having combinations of consonants and vowels with punctuation marks, a space, or a digit[38] as a delimiter. This task is accomplished using a function written using an algorithm presented in Figure 4.2. The algorithm requires a buffer area to store a word read.

[38] Words used in our every day communication do not include digits as their parts. Thus an existence of a digit in a corpus is considered as the end of one word and the beginning of the other.

1. If end of file is reached, stop else do
    a. While a consonant or a vowel is not found, read a character from the file
    b. If a character is found, do
        i. Put the character into a buffer
        ii. Read a character
        Until a space, punctuation mark, a digit or end of file is encountered
    c. Pass the buffer to the function that extract the morpheme component
2. Return to step 1

Figure 4-2: An algorithm to read a word from a file

Actually what is referred as a word so far is a stem. And for autosegmental representation of the stems, each stem read from the input corpus needs to be analyzed. That is to say, the consonantal root, vocalic melody and the organizing template of a stem should be identified. The algorithm presented in Figure 4.3 is designed to handle this task.

1. get a stem
2. read a character from the stem
3. If it is a vowel,
    a. Check the existence of the character in the previously identified vowels of the word.
        i. If it is found, add the character to the **TEMPLATE**;
        ii. Else if it is not found, append the character to the **VOCALIC,** and add the character to the **TEMPLATE**;
4. else (if the character is a consonant)
    a. Check the existence of the character in the previously identified consonants of the word.
        i. If it is found, take the position of the character and add it to the **TEMPLATE**;
        ii. Else if not found, append the character to the **CONSONANT,** and take the position of word and add it to the **TEMPLATE**;
5. if not end of a word, go to step 2
6. pass **TEMPLATE, VOCALIC, CONSONANT** to
    a. a function that construct the different tiers of an autosegmental representation
    b. a function that construct the association tier (called here a stem signature)

Figure 4-3: Algorithm for extracting morphemic component of stems

As already discussed in chapter 2 and 3, Semitic stems involve a number of morphological processes internal to their stems. These processes can be categorized into radical extension –

that is extending one or more of radical in the roots, and radical reduction – removing one or more of their radical during stem formation. Radical reduction involves consideration of phonological properties of each phoneme (characters read in our case) and it is out of the scope of this research. But root extension is considered. Semitic stems undertake two types of root extension – gemination as in /säbbärä/ or reduplication as in /asäbabbarä/. Moreover, some consonants not included in the root can also be added during stem formation as in /t↗säbri/. The algorithm in Figure 4.3 considers all these complexities involved in Semitic stem formation.

The algorithm accepts a stem as an input. Then it declares three spaces to store the consonantal roots, vocalic melodies and stem template identified. It reads a character, checks whether it is a consonant or a vowel. Before putting the character in the respective spaces, it checks the existence of the character read in character sets stored previously. The algorithm adds the character only if it is not stored already. This is used to avoid the impact of extended consonants in root/vocalic melodies identification. This task is accomplished by instruction at 3.a.ii and 4.a.ii of Figure 4.3.

However, these extended characters should remain as part of the stem template as those observed in the skeletal tiers of autosegmental phonology (see Figure 3.1). As stem templates can be used by a number of words, symbols are used in place of actual characters in the template representation.

There are two types of stem template representations in the literature. The first puts variables such as V_1,V_2 to represent a place for infixing vocalic melodies while the second approach put the actual vowels within the template. For example, /Säbbärä/ can be represented as

$C_1V_1C_2C_2V_1C_3V_1$ in the first case and as 1ä22ä3ä in the second case, representing consonants by numeric sequence (such as 1, 2, 3…). Since it is found linguistically advantageous[39], the second approach is selected. The numeric variables designate the position of a character in a consonantal root tier (they serves as Indexes). For example, /säbbärä/ has a root {sbr} with [b] as its second radical. This task is performed by the instruction depicted in 3.a.i and 4.a.i of Figure 4.3.

The last part of Figure 4.3 (6.a and 6.b) passes the extracted consonantal root, vocalic melody, or stem template (referred as morphemic components hereafter) to functions that construct different tiers of the autosegmental representation and the linker stem signature. The algorithm to construct different tiers of the autosegmental representation is presented in Figure 4.6, and the one to create the stem signature is put in Figure 4.7.

However, as it is discussed in detail in the next chapter, some problems are encountered during the experiment, which needed modification of the algorithm presented in Figure 4.3. During the experimentation, a stem with identical radicals in different position as in /lella/ 'another', /t'ät't'a/ 'drank' and /märämmärä/ 'examine' are identified as {l}, {t'} and {mr}. These are incorrect. Figure 4.4 gives an algorithmic solution for such problems.

The modification made is presented in step 5.iii and 7 of Figure 4.4. These components are added to deal with bi-radical consonant with identical radicals. The instruction at 5.iii increments the counter, cnt_radical, by 1 whenever two identical consonants don't come together. This information is used by the code at 7 to modify the content of CONSONANT.

[39] Tested the two, the first reduces number of CV template required to represent stems in the corpus, but it brings stems with different vocalic melodies into one group. As a result the 2nd method is selected.

1. declare an integer variable **cnt_radical**, and a consonant variable **PrvLttr**
2. get a word
3. read a character
4. If it is a vowel,
    c. Check the existence of the character in the previously identified vowels of the word.
        i. If it is found, add the character to the **TEMPLATE**;
        ii. Else if it is not found, append the character to the **VOCALIC,** and add the character to the **TEMPLATE**;
5. else if the character is a consonant
    a. Check the existence of the character in the previously identified consonants of the word.
        i. If it is found, take the position of the character and add it to the **TEMPLATE**;
        ii. Else if not found, append the character to the **CONSONANT,** and take the position of word and add it to the **TEMPLATE**;
        iii. If the character is not the same as **PtrLttr,** add 1 to **cnt_radical**
6. if not end of a word, go to step 2
7. if **cnt_radical** is 2 and length of **CONSONAT** is 1; duplicate content of **CONSONANT** once
8. pass **TEMPLATE, VOCALIC, CONSONANT** to
    a. a function that construct the different tiers of an autosegmental representation
    b. a function that construct the autosegmental representation (called here a stem signature)

Figure 4-4: Algorithm for extracting morphemic components of stems (modified)

1. accept a consonant root, vocalic melody and template
2. search the respective tiers for root, vocalic melody or template;
3. if they are found, add 1 to their frequency
4. else, get a space for root, vocalic melody and template in their respective tiers
5. copy the content of the root, vocalic melody and template into the newly assigned space in the tiers.

Figure 4-5: Algorithm for Constructing Autosegmental Tiers

Similarly, root and vocalic melody tiers are created as follows. As depicted in Figure 4.5, a morphemic component is accepted as an input from a function that uses the algorithm depicted in figure 4.3. Then the input morphemic component are searched in their respective data structure (tiers) (the search algorithm used is presented in Figure 4.8). If the sought

morphemic component is found, its frequency is incremented[40]. But if it not found, the morphemic component accepted is considered as a new and added to the data structure. Addition of new morpheme or template into a data structure demands assignment of new memory space, which can easily be done using the inbuilt function available in most programming languages.

However, putting morphemes into different tiers without showing the relationship that exist between them do nothing good for the language. Thus, a structure showing this relationship (called an association in the autosegmental phonology) should be established. Figure 4.6 gives the algorithm that does the association between autosegmental tiers.

1. accept a root, vocalic melody and a template
2. search the stem signature for a template
    a. if found, do ADD_ROOT and ADD_VOCALIC tasks
    b. else if not found
        i. get a space in the stem signature
        ii. store address of a template in a TEMPLATE data structure into the stem signature
        iii. do ADD_ROOT and ADD_VOCALIC tasks

Figure 4.6 presents an algorithm for creation of a linking structure that is used to show the relationship that exists between different tiers in autosegmental representation. The algorithm and the associated data structure are designed taking into consideration the importance of stem templates in distinguishing the category of verbal stems. As discussed in chapter 3, for example, /CäCCäC/ is a perfective verbal stems, while /CäCC-/ represents an imperfective whereas /CCäC-/ shows jussive verbal stems. Depending on the template used, a consonantal root can be a perfective, imperfect, jussive or any other kind of stem.

[40] This frequency information is useful to know the frequency of a particular root, vocalic melody or a template in a give corpus, and thereby in the language.

With this conception, the algorithm in Figure 4.6 creates the stem signature, by storing the address of a particular template together with addresses of consonantal roots and vocalic melodies that use a particular stem template. This is done in the following way. The algorithm accepts input morphemic components; then the stem signature is searched for a template (using the modified version of the search algorithm at Figure 4.8). Two different operations are preformed depending on the search results.

If the template searched is not found, the template is considered new to the signature. Thus the address of the template is added together with the address of the root and vocalic melodies. The address of the morphemic components is added by searching the respective tiers using the algorithm at Figure 4.8. But if the template is found, since the template already exists in the stem template, the existence of the addresses of the root and vocalic melody are checked and added if they are not found (Figure 4.7, which is a modification of Figure 4.5, is used for this pu[illegible])

1. get a root/ vocalic melody
2. search the consonantal root tier or vocalic tier for the root/vocalic melody
3. if the root/vocalic melody is not found
    a. get a space
    b. copy the content of root/vocalic melody into the space.

Figure 4-7: An algorithm to add addresses of morphemes to a Stem Signature

1. Accept as input the autosegmental tier to be searched and the item to be searched
2. do
    a. if end of file is reached RETURN NULL
    b. else if the item is found RETURN address of the item
    c. else read the next record
    d. go to 2.a
3. end;

Figure 4-8: An Algorithm to search autosegmental tires

The algorithm in Figure 4.8 is used to search for particular morphemic components in an autosegmental tiers. The algorithm accepts an input autosegmental tier name and a morphemic component to be searched. The algorithm then search the tier until either the item searched is found or the end of file is encountered. The algorithm returns the address of the morphemic component if it is found or NULL otherwise.

4.3 Data Structures

This section discusses data structures designed for use in the storage of autosegmental tiers and stem signatures. The type of data structure used is a binary search tree. Binary search tree is selected because of its better performance in insertion and searching. Since this work is an experimental one, the system is a memory-based, which make binary-search preferable data structure. In the case of real-life application, however, since the size of the different tiers become larger and larger with inclusions of more records, file-based structures such as B-Tree or B+Tree are required.

Two basic data structures are designed. The first is used for the autosegmental tiers and the other for the stem signature. But the stem signature needs an internal structure to store list of roots and vocalic melodies associated with a particular template. Figure 4.9 shows the data structure used to store different autosegmental tiers[41], while the internal data structures used for the stem signature are presented in Figure 4.10 and Figure 4.11.

[41] a character pointer '*gloss' is added to store root glosses and to label the stem templates

```
typedef struct Morpheme{
        char *morph;
        int freq, biradic, triradic, quadriradic;
        int quiniradic, sixiradic;
        struct Morpheme *left, *right;
        } morpheme;
```

Figure 4-9: Data Structure of the Autosegmental Tiers

As it can be observed from Figure 4.9, the structure 'morpheme' has three pointers '*morph', '*left' and '*right' in addition to integer variables that are used to store frequency information. The pointers '*left' and '*right' are used in the binary tree organization. The '*morph', on the other hand, is used to store the consonantal roots, vocalic melodies and stem template in their respective tiers. It is declared as pointer to use memory spaces wisely by allocating memory dynamically at run time. The algorithm that uses this data structure to allocate memory space is designed to terminate the program when the system runs out of memory.

```
typedef struct StemSignatureStructure{
                        morpheme *ptrtmp;
                        Associoation *ptrvm;
                        Associoation *ptrrt;
                        struct StemSignatureStructure *left, *right;
                        } StemSignatureStr;
```

The algorithm in Figure 4.10 is the one designed to store stem signatures. It has a pointer '*ptrtmp' of type morpheme (described above), and other two pointers '*ptrvm' and '*ptrrt' of type association. The pointer '*ptrtmp' is used to locate to a memory position where a template is stored. The other two pointers are used as a root node of internal trees that are used to store lists of consonantal roots and vocalic melodies associated with a particular template. The structure of these internal trees is presented in Figure 4.11.

```
typedef struct associoation{
                    morpheme *ptr;
                    struct associoation *left, *right;
                }Associoation;
```

Figure 4-11: Data structure to represent morpheme associations

To summarize, this chapter discussed the corpus prepared, and the algorithm and data structures designed for the autosegmental representation of Amharic stems. Algorithms that are used to extract words from a corpus file, extract the stem morphemic components and store them in different tiers are presented. The algorithm used to produce a stem signature and the data structures used for the autosegmental tiers and stem signatures are also discussed. Based on the algorithms, an Amharic stem morphological analysis system, named ASMA is developed.

For the purpose of training and test, two types of corpuses are also prepared – one for a test using Linguistica2001 and the other with Amharic stem morphological system, ASMA. Two separate experiments are also conducted using these corpuses. The experiment, the experimental procedures and the final experimental result are presented in the next chapter.

CHAPTER V

The Experiment

This chapter discusses the experiment conducted using Linguistica2001 and ASMA. In the discussion emphasis is given to assess the outputs produced and the test result found. Coverage is also given to the test data and testing method used in the experiment. The results of the tests are presented in **percentage**, and in **precision** and **recall** ratios.

5.1 Experiment with Linguistica2001

Linguistica2001 is used to **identify the concatenative morphology of Amharic words**, i.e., the prefix, stem and suffix components of words in a given corpus. Using these morphemic components identified, the system also produces a morphological dictionary, referred to as **signature.** This dictionary is designed to serve as a means to show permissible word formation patterns (Goldsmith, 2001a).

The input to Linguistica2001 is a text file containing a natural language word lists or sentences of any sort. Goldsmith (2001a) reported that his system was tested with corpuses as large as 124,726 Spanish words, 125,000 Latin words, and 100, 000 & 1,000,000 Italian words. However, the current experiment is conducted with **5,236-word size corpus**, which is the maximum the researcher was able to prepare within the existing time and financial constraints. Though the size of the corpus used is incomparably very small, it fulfils the least size recommended by Goldsmith (2001a), i.e., a corpus size of 5,000 words.

Though the effect of the small corpus size used is quit observable in some of the outputs, the overall result seems encouraging. The system has identified prefixes and suffixes of the language in significant number. Detailed description is presented in section 5.1.2, next to a brief explanation on the experimental environment in the following section.

5.1.1 The Test/Experiment Environment

This experiment is conducted using various modules of Linguisitica2001[42]. The first module applied (labeled Successor Freq 1) uses a heuristic based on the modified version of Harris's Algorithm of Successor Frequency (see the description in section 2.4.2.1). The module is used to provide a baseline morphological analysis. Output found by applying this module is then refined using other modules (labeled **Known Stems and Suffixes**, **From Signatures find stems**, **Loose fits, Check Signatures, and Smooth Stems**.) These modules use stems, prefixes and suffixes identified by Successor Freq 1 to identify others which are not recognized before. Moreover, the quality of the stems, prefixes, and suffixes identified is improved by removing part of them or attaching part of one onto the other.

Linguistica2001 has also a module to enable users set threshold values such as minimum lengths (of a stem, a suffix , a prefix, and signatures), minimum number of stems and affixes in a signature and minimum successor frequency desired. Table 5.1 lists these thresholds set.

[42] Detailed description of Linguistica2001 and its modules is found at ***http://humanities.uchicago.edu/faculty/goldsmith***

Threshold Specifications	Threshold	Description
Minimum successor frequency	6	based on test result
Minimum Stem Length	3	observing stems of the language
Minimum Signature Length	1	the default accepted
Minimum Number of Stems in the Signature	1	the default accepted
Minimum Number of Character to save to keep signature in triage processes	15	hard wired in the program

Table 5-1: Thresholds Used in the experiment

In this experiment the minimum length of a stem is set to 3, because of the existence of stems like {säw} 'man', {set}, 'woman' and {bet} 'house'. Moreover, the minimum successor frequency is set to 6, after a test with 3, 4, 5, and 6 threshold values. Actually, the number of signatures produced with the minimum successor frequency of 3 and 4 are large in number but the majority of them are erroneous, and setting it to 5 or 6 brings no difference. As a result minimum successor frequency of 6 (which is the default) is used in the experiment. Finally, for the minimum stem and suffix numbers required to keep a signatures are set to 1 taking into consideration the small size of corpus used.

5.1.2 Initial Inspection of Amharic Signatures

As stated above, the experiment is conducted with a corpus of 5,236 words. Linguistica2001 identified 3,331 distinct words from this corpus. This implies that the majority of words in the corpus appear once or twice. Since Linguistica2001 analyses words based on the number of occurrences of a given sequence of letters in the corpus, the existence of such low frequency words had a negative effect on the quality of the result (see section 2.3.2.1 of Chapter 2). Similar effect observed in this experiment is presented later in this section.

Application of various modules of Linguistica2001 produced an output that seems good from onset. In the first run 119 suffix signatures, 16 prefix signatures, 34 suffixes, 15 prefixes, and 49 compound words were found. Moreover, as suffixes and prefixes are analyzed separately, 321

stems associated with suffix signatures, and 86 stems associated with the prefix signatures were identified. Table 5.2 and Table 5.3 present the prefix and suffix signatures respectively.

Rank	Signature	# Stem	Rank	Signature	# Stem	Rank	Signature	# Stem
1	NULL.yä	21	7	sl.y	3	13	b.l	2
2	NULL.bä	14	8	NULL.k	3	14	NULL.tä	2
3	NULL.kä	10	9	NULL.as	3	15	NULL.lämä	2
4	NULL.lä	8	10	b.k	2	16	NULL.m	2
5	b.y	6	11	k.l	2			
6	k.y	4	12	la.lä	2			

Table 5-2: Prefix Signature Identified

The suffix and prefix signatures presented in Tables 5.2 & 5.3 contain common suffixes, prefixes, prepositions and clitics of the language. For example, the top seven suffix signatures found, NULL.u, NULL.w, NULL.n, NULL.ïn, NULL.u.un, NULL.un, o.äw, and NULL.nna, contain the following suffixes NULL, {u}, {w} {n}/{ïn}[43], {o}, and {äw}. Depending on the type of words they are attached to, these suffixes can be used in a number of grammatical contexts. For example, {u} is used in /lïjju/ 'the boy' as a determiner, but in 'addisu bet**u'** 'his new house' the suffix {u} is used in the second word as a possessive.

The fifteen prefix signatures identified are also commonly observable morphs of the language. Amharic has small number of prepositions that are used in many different ways (Baye, 1994). Thus they might be found in sufficiently large number even in a small corpus size like the one used in this experiment. It seems as a result of this that the majority of the prefixes identified are prepositions like {kä}, {lä}, {yä}, {slä} and their assimilated forms {k}, {l}, {sl} and {y} (see Table 5.2). Besides these prepositions, other type of prefixes such as the causative {as-} and the passive maker {tä-} are also identified.

[43] {n} and {ïn} are the same, they are presented as different because of the existence of [ï] inserted to separate impermissible consonants (see chapter 3)

Rank	Signature	# Stems	Rank	Signature	# Stems	Rank	Signature	# Stems
1	NULL.u	27	41	o.uall.ä.äw	1	81	u	4
2	NULL.w	21	42	a.o.uall.äw	1	82	äw	4
3	NULL.n	18	43	NULL.un.ïm.ïn	1	83	um.ïm	1
4	NULL.ïn	16	44	NULL.e.u.ïm	1	84	qä.rä	1
5	NULL.u.un	6	45	NULL.m.n.w	1	85	l.mä	1
6	NULL.un	13	46	NULL.occ.u.un	1	86	o.uall	1
7	o.äw	10	47	NULL.a.m.n	1	87	a.ä	1
8	NULL.nna	9	48	NULL.u.um.un	1	88	u.äwïn	1
9	NULL.wïn	8	49	NULL.s.w.wïn	1	89	h.w	1
10	NULL.m	7	50	NULL.e.u.ïn	1	90	l.u	1
11	ut.ä.äw	3	51	NULL.n.w.wïn	1	91	NULL.e	1
12	NULL.w.wïn	3	52	NULL.u.un.ïm	1	92	ut.äwïn	1
13	NULL.all	6	53	NULL.u.un.ïn	1	93	occu.u	1
14	NULL.a	6	54	NULL.u.um	1	94	NN.h	1
15	NULL.occ	6	55	NULL.occ.ïm	1	95	all.ä	1
16	NULL.ccäw.w.wïn	2	56	e.u.um	1	96	o.ä	1
17	NULL.occ.u.wa	2	57	NULL.bät.w	1	97	e.u	1
18	ut.äw	5	58	NULL.m.n	1	98	all.äwïn	1
19	NULL.ïm	5	59	NULL.w.äw	1	99	bät.l	1
20	a.äw	4	60	NULL.nna.w	1	100	NULL.ccäw	1
21	NULL.wa	4	61	NULL.m.w	1	101	NULL.h	1
22	NULL.äw	4	62	NULL.a.n	1	102	ïm	3
23	NULL.n.nna	2	63	NULL.m.s	1	103	un	3
24	NULL.l.u	2	64	bät.w.wïn	1	104	m	2
25	NULL.occ.u	2	65	a.o.äw	1	105	rä	2
26	NULL.u.ïm	2	66	a.u.ä	1	106	ä	2
27	NULL.e.u	2	67	NULL.occu.u	1	107	occu	2
28	NULL.ut.äw	2	68	NULL.u.ua	1	108	w	2
29	NULL.äw.ïm	2	69	NULL.e.ïn	1	109	NN	2
30	NULL.e.occ.u.un.wa.ä.	ïm 1	70	NULL.wa.wïn	1	110	s	2
31	u.ä	3	71	NULL.n.w	1	111	äwïn	1
32	NULL.um	3	72	NULL.h.w	1	112	um	1
33	NULL.bät.m.nna.w	1	73	NULL.ïm.ïn	1	113	h	1
34	NULL.u.ua.un.ïm	1	74	NULL.u.ä	1	114	qä	1
35	NULL.m.s.un.ïm	1	75	NULL.un.ïn	1	115	e	1
36	NULL.a.occ.u.ïm	1	76	NULL.all.u	1	116	mä	1
37	a.u	2	77	NULL.occu.ïn	1	117	all	1
38	NULL.unna	2	78	NULL.occu.ïm	1	118	bät	1
39	NULL.s	2	79	o	6	119	l	1
40	NULL.u.ua.un	1	80	a	4			

Table 5-3: Suffix Signatures identified in the first run

The signatures represented in Table 5.2 and 5.3 are used as links between affixes and stems. They show the affix use pattern of the stem component. In the Linguistica2001 interface, clicking one of the signatures displays all stems that use the suffix/prefixes of the signature. Figure 5.1 shows an example.

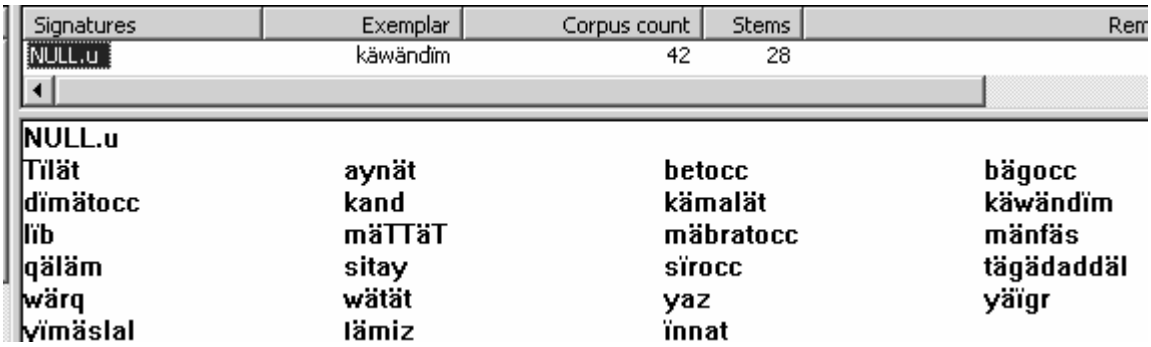


Figure 5-1: An example of Stems appearing with a particular signature

Figure 5.1 shows all stems that appear with a **NULL.u** signature, i.e., stems that are found with the NULL and {u} suffixes. The figure, for instance, shows that NULL.u appears with {lïb} 'heart', {wätät} 'milk' and {ïnnat} 'mother'. These words can be used adding the suffix {u} as in /ïnnatu/ 'his mother' and /wätätu/ 'the milk' or without adding any suffix as in {wätät} 'milk' and {lïb} 'heart'. In such cases, the words are considered to appear with a NULL Suffix (see section 1.1 of Chapter 1).

However, there are some problems in the stems displayed. Among the stems displayed, /dïmätocc/ 'cats', /betocc/ 'houses', /bägocc/ 'sheep (pl.)' have suffix remnant {-occ}, the plural maker, which should have been removed out from the stem part by the system. Close observation of the entire signature also reveals that there are such erroneous results. The problems identified can be summarized in the following way: -

1. Inclusion of prefixes and suffixes (or part of them) as part of the stem components: like {-w} of */afn◌̃c'aw/ 'his* nose' and {y◌̃-} & {-al} of */y◌̃-noral/* 'he lives'.
2. Merging of two suffixes into one: for example, we find the plural maker {-occ} and the determiner /-u/ collapsed into /-occu/. Similar spurious suffixes found are /-wïn/, /-um/ and /-un/. For example, /-wïn/ is collapsed from the possessive marker {-w} and the determiner {-ïn} as in /jorro-w-ïn qorrät'-ä/ [ear-his-the cut he] 'he cut his ear'.
3. Inclusion of spurious signatures. Such a result is found with a regular suffix {-n}. Though {-n} is a real morpheme, it is used erroneously as a suffix of /◌̃nk⊕a/ which is incorrect. The correct stem is /◌̃nk⊕an/. The error seems to happen due to the appearance of [n] in word-final position, which created difficulty to distinguish it from the regular suffix {-n} as both of them appear in word-final positions.
4. Consideration of grammatical variations of the same stem/suffix as completely different stem/suffix. For example, {-n} in the 3rd signature *NULL.n* and {-ïn} in the 4th signature *NULL.ïn* are the same except the deletion of [ï] in the former signature due to assimilatory process.

To correct such problems, Goldsmith (2001a) uses some heuristics that remove some signatures and modifies the remaining signatures. An MDL test is also used to accept/reject the modification. The first activity in this line is removing all signatures that either appears with only one stem or those that have only one suffix of one character long. Goldsmith (2001a) considered such signatures as spurious signatures. To detect them, the **log (stem count) * log (affix count)** (described in Section 2.3.2.2) is computed for each signatures identified. This log value is then used as a rough guide to the centrality of a signature in the corpus, and those signatures with zero log values are removed.

Furthermore, using the **Check Signature** and **Smooth Stem** modules of Linguistica2001, an attempt was made to correct some erroneous segmentation. Application of these modules helped to refine the stems, and suffixes identified by initial segmentation. For example, the

initial segmentation parses /t'ägurun/ into a stem /t'äguru/ 'his hair' and a suffix {n}. But /t'äguru/ still has two morphemes – {t'ägur} 'hair' and {-u} 3m.sg.POS. These components were identified by the application of Smooth Stem.

However, the MDL test is not conducted in the experiment because of a problem in Linguistica2001[44]. Despite this, the application of the remaining modules removed 71 of the signatures appearing only with one stem and another 14 signatures found with only 1 suffix of length 1 character. After that, only 34 signatures were remained. In Goldsmith's terminology, the set of signatures remained are referred to as **regular signatures**. The suffixes/prefixes in these signatures are also referred to as **regular affixes**. Table 5.4 lists the regular signatures identified.

Rank	Signature	# Stems	Rank	Signature	# Stems	Rank	Signature	# Stems
1	NULL.u	27	13	NULL.all	6	25	NULL.occ.u	2
2	NULL.w	21	14	NULL.a	6	26	NULL.u.ïm	2
3	NULL.n	18	15	NULL.occ	6	27	NULL.e.u	2
4	NULL.ïn	16	16	NULL.ccäw.w.wïn	2	28	NULL.ut.äw	2
5	NULL.u.un	6	17	NULL.occ.u.wa	2	29	NULL.äw.ïm	2
6	NULL.un	13	18	ut.äw	5	30	u.ä	3
7	o.äw	10	19	NULL.ïm	5	31	NULL.um	3
8	NULL.nna	9	20	a.äw	4	32	a.u	2
9	NULL.wïn	8	21	NULL.wa	4	33	NULL.unna	2
10	NULL.m	7	22	NULL.äw	4	34	NULL.s	2
11	ut.ä.äw	3	23	NULL.n.nna	2			
12	NULL.w.wïn	3	24	NULL.l.u	2			

Table 5-4: Amharic Signatures Identified

Nevertheless, a number of signatures removed in this process were observed to be important signatures of the language. One of the useful signatures missed was ***NULL.e.occ.u.un.wa.ä.ïm*** that appears with /bet/ 'house' (See Table 5.3). The signature holds suffixes such as 1p.sg, 3m.sg. and 3f.sg. possessive pronouns {e}, {u} and

[44] From email contact with the developed, it is known lately that the MDL module is under development.

{wa}. Another important suffix the signature holds is /-ä-/ which is Ge'ez[45] compound maker as in /bet-ä-kïrstiyan/ 'church'.

Despite such losses (presumably due to small corpus size) used, the application of these sets of heuristics removes a number of erroneous signatures. For example, the signature *qä.rä* , which is ranked 84th, is derived from /märräqä/ 'bless' and /märrärä/ 'become sour' considering /märr/ as their common stem, which is incorrect.

However, the majority of the words in the corpus are not analyzed. As indicated above only 338 stems associated with suffixes, and 78 stems associated with the prefixes are identified. Compared with 3,331 distinct words identified, the number of stems identified is too small. Moreover, some stems identified such as /dïmätocc/ 'cats' have additional components like {-occ} 'pl.' which ought to have been parsed out to produce linguistically accepted stems.

Linguistica2001 has also a module **Known Stems and Suffixes** which identifies morphemic components from unanalyzed words based on stems and suffixes already identified. But application of this module did not bring much difference in the result.

5.1.3 The Test with Linguistica2001

In the preceding section, the signatures identified by Linguistica2001 are examined. A test is also conducted to **examine and quantify how well words of the corpus are analyzed morphologically.** The test was carried out using 500 words selected from the corpus (about 1/6th of the unique words in the corpus). These words are chosen simply by taking the first 500 words from alphabetized list of the analyzed words (words segmented in some way by the systems).

[45] Geez is a language from which Amharic took much of its structures.

The test is conducted following the method employed by Goldsmith. In this method, words are put into one of the four categories, namely, **good**, **wrong analysis**, **Failed to Analysis**, and **spurious analysis** depending on the quality of their parses (i.e., the correctness of the stem and affix components of the words identified). A word analysis is considered correct (put in good category) if and only if all its parses are correct. Else if an analysis of a multi-morphemic word is attempted but not correct, the word is grouped in the "Wrong analysis", whereas "Failed to analyze" is used for multi-morphemic words for which no analysis is provided. Lastly, the "spurious analysis" comprises single-morph words such as {säw} 'man' which is wrongly analyzed as containing a suffix or a prefix.

However, the "**good**" category is used with some modification. It is shown in the previous section that some of the stems identified have additional morphemic components. Assuming that the existence of such multi-morphemic stems is due to the small size corpus used, words having such stem are considered correct as far as they are parsed correctly. For example, if the system parses /säwoccu/ 'the men' into a stem /säwocc/ 'men' and the definite article {u}, the word parse is considered as correct even though the plural maker {occ} remains on the stem.

With these modifications, the test data was given to 2 linguists (instructors at the Linguistics Department, AAU) to categorize them based on the above criteria. The test result shows that 403 words (86%) of the 500 test corpus were analyzed correctly (are in good category). But the system gave spurious analysis for 20 words and failed to analyze 46 words and wrongly analyze one word.

Category	Count	Percentage
Good	433	86.6%
Wrong analysis	1	0.2%
Failed to analyze	46	9.2%
Spurious analysis	20	4.0%

Table 5-5: Test Result

On the other hand, the system's performance can be presented in precision/recall ratios as used in Goldsmith (2001a). Considering precision as a ratio of correct analysis to total number of words to which an analysis is attempted, and recall as a ratio of total number of words analyzed correctly to total number of analyzable words (words with 2 or more morphemic components), the system has a precision of $\left(\left[\frac{433}{433+1+20}\right]*100\right)$ = 95.37% and a recall of $\left(\left[\frac{433}{433+1+46}\right]*100\right)$ = 90.21%.

5.2 Experiment with the prototype Stem Analyzer

The pervious section described the experiment carried out on Linguisitca2001. The main focus there was developing signatures that would serve as a bond between prefix, stem and suffixes. However, as discussed in Chapter 1 & 3, word stems in Semitic languages, including Amharic, have analyzable components, i.e., consonantal root, vocalic melody and stem template components. The general pattern of stem formation in Amharic involves infixing vocalic melodies into consonantal roots following patterns in stem templates.

However, Linguistica2001 cannot handle such processes (see Section 2.6). As a result, a separate prototype system is developed as part of this study to handle the stem internal morphological analysis task. The system developed, called **ASMA,** analyzes Amharic stems into **consonantal root, Vocalic Melody** and **stem template** components**.** It puts these

components in different tiers and maintains a structure, labeled **stem-signature**, to retain the relationships that exist between them. .

In the current implementation, the input of the system is a text file-containing list of stems. Analyzing the input stems, the prototype system produces a stem-signature as an output. However, because the majority of the stem components identified by Linguistica2001 have additional non-stem components, they are not used for this test. Rather a corpus containing 326 stems is used in the development of the stem-signature and to test the system. The following two sections describe the stem signature developed (in section 5.2.1), and the test conducted (in section 5.2.2).

5.2.1 The Stem Signature

Stem templates are useful to form stems from consonantal roots and vocalic melodies (see section 3.6). They tell us in which tense, aspect or mode a word is used. For example, forming a stem from a consonantal root {sbr} using CVCCVC- and –CVCC- gives two grammatically different forms of the same root. Using CVCCVC- we form the perfective /säbbär/ 'broke' while with –CVCC- we form the imperfective like /lisäbr/ 'to break'.

Stem templates are also considered as a morpheme component. They keep a separate tier in the autosegmental representation. However, having simple list of stem templates in a separate tier might not help much for real life morphological applications. A system must be maintained to tell the interrelationships that exist among different components stored in different autosegmental tiers.

The prototype system developed has such a module. The module creates a structure called **stem signature** which associates each roots and vocalic melodies to a template with which they are found in the corpus. Technically, the stem signature is a list of pointers to the template, consonantal root and vocalic melody tiers that hold the actual morphemic components. Figure 5.2 presents a sample Stem Signature (See ***Appendix 2*** for Screen snapshot).

```
1ä22ä3ä    ä
                    šmt
                    škr šbt znb wsd wrd wfr wTr trb tmn tlq
                    tkz tkl srq skr rzm rTb qzn qtr qst qrm
                    qrf qlT qbr nql nTl mrq mrg mqs mlg lws
                    lsn ktm ksm ksl kmr kbd jmr grz grd gdm
                    frs flq flT drq btn brq Trb Tbq Sly Clm
                    CbT
1ä2a22ä3    äa

                    sbr lbs gdl
11ä2a22ä3   äa
                    gdl
1ä23    ä
                    wnd sbr qst qnd
1ä2ä3    ä
                    tmn snf sbr mqs kft grd frs dnz
```

Figure 5-2: Sample Stem Signature

The stem signature depicted in Figure 5.2 shows that some 52 consonantal roots form their stem using the perfective template 1ä22ä3ä, which is the largest stem signature produced. The figure also shows templates of derived stems such as the frequentative 1ä2a22ä3, the adjutatuve (addaragi) 11ä2a22ä3, the gerundive 1ä23 and the noun stem template 1ä2ä3 together with the associated roots and vocalic melodies.

From the observation of the stem template, consonantal root and vocalic melody components of the signatures, the results produced seem linguistically acceptable. Both a simple trilateral as well as derived stems such as the frequentative / gädaddäl / 'kills more and more' are properly analyzed

into consonantal root {gdl} and a template {1ä2a22ä3}. Moreover, some quinqui -radical and sexi-radical stems are also analyzed into tri-radical roots as argued in Baye (1999)[46]. Examples are analysis of /ašfädäffäd/ to {šfd} and /ažguädäguäd/ to {žgd}.

However, this could not substitute the importance of testing to measure the effectiveness of system. Hence, the prototype system developed was tested with a corpus of 255 stems. This test is discussed in the following section.

5.2.2 The Test with ASMA

In the test, the quality of a stem analysis is evaluated by examining the consonantal root, stem template and vocalic melodies produced for each stem. A stem parse is considered correct (or good) only if all its stem components were extracted correctly, else it was considered as wrong analysis.

However, a problem is encountered in the analysis of mono-radical and bi-radical words such as /ša/ 'desire' and /bälla/ 'eat'; and those stems with dropped radicals like /awwäkä/ 'know' and /ⷪnnat/ 'mother'. In the literature, stems showing these features are believed to undertake radical reduction (see Chapter 3). For example, the stem /färra/ 'feared' is believed to drive from a consonantal root {frh} dropping its ultimate (last) radical (Baye, 1999). In theoretical linguistics, [h] is assumed to get dropped during stem formation. Thus, it is added to the

[46] Baye (1999) argues that Amharic has only tri-radical roots; considering all others derivations of some sort.

component /fr/ which is identified from /färra/ to form the morpheme component {frh}. But these processes are purely phonological[47] and are not considered in this study.

Because of the problem, stems with missed radical are not considered in the test (some 71 stems of the corpus are excluded). The test is thus conducted using 255 stems of the corpus. The test result shows that the system has managed to identify the consonantal roots of 241 stems correctly, whereas the stem template and vocalic melodies of all the stems are identified correctly. The result is presented in Table 5.7.

Stem Component	Correct Analysis	Percentage
Consonantal Root	241	94.51%
Template	255	100.00%
Vocalic Melodies	255	100.00%
Total	255	

Table 5-6: Result of Stem Component Analysis

The errors found are in quadric-radical stems with identical sequence of characters repeated one or more times as in /märämmärä/ 'investigate' and /säbässäba/ 'collect'. The analyzed roots of these stems given are {mr} and {sb} which are wrong. Their correct analyses are /mrmr/ and /sbsb/. Such errors are encountered because the system ignores duplicated characters during the analysis to avoid extended characters in such stems as /gädaddälä/. Though this is not done here due to time limitation, this problem can be easily corrected by

[47] Though it is not tested in this study, this problem can be handled by a probabilistic approach that uses ***two-level phonology*** (that is, phonology with underlying forms and surface forms). Such a model contains two phonological representations (one underlying, the other surface), with correspondences between elements of the two levels. Log probability of each of these links can be computed from a given corpus, and this log information can be used as a component for morphological analysis. Further discussion is available in Goldsmith (2001b) and Albright & Hayes (1999).

modifying Step 7 of the algorithm presented in Figure 4.4 to incorporate all stem types of the language.

All in all, the experimental result found seems satisfactory. Though the two systems, Linguistica2001 and ASMA, are tested separately using two separate corpuses in this study, they can be used in an integrated manner. Had it been tested with large corpus, the stem component identified by Linguistica2001 might be good enough to be used as an input for the prototype stem analyzer. In the long-run, however, the stem analyzer developed should be integrated with Linguistica2001. The integration is required to maintain the relationships that exist between prefixes and suffixes extracted by Linguistica2001 with the stem components extracted by ASMA. Further discussion on these issues is presented in the following chapter.

CHAPTER VI

Conclusion and Recommendations

6.1 Conclusion

This thesis reports on an attempt made to develop Amharic morphological analysis system employing a combination of unsupervised learning and autosegmental analysis approaches.

The report started off with brief introduction to concepts and principles used in the study. The introduction also includes description of the unique feature of Semitic words along with their peculiar morphemic components. The role of morphological analyzer in natural language understanding systems is also discussed. Moreover, an attempt is also made to review some works conducted in the area of NLP system development for Amharic.

Review of different approaches used in development of morphological analysis system is presented in Chapter 2. The approaches found are broadly categorized into rule-based and corpus-based approaches. The two subcategories of corpus-based approaches, supervised and unsupervised, are described focusing on their advantages and disadvantages. In this study, unsupervised approach was considered for the reasons stated in chapter 2.

Giving brief account on Amharic phonemes and word classes, chapter 3 described the morphological complexity of Amharic words. Word formation in Amharic involves three phases – **stem formation** from consonantal roots; **inflection**, and **cliticization**. There are also additional morphological and phonological processes involved in each phases of word formation.

But the inflectional and cliticization processes are similar with those in Indio-European languages like English and French. Hence rather than starting from scratch, a **language-independent system** tested in these languages is used to handle morphological analysis involving these two processes. The system, called **Linguistica2001,** analyzes words applying a series of heuristics. The first heuristic segments words into promising parses studying word patterns in a given corpus. After making some modification on the output and constructing a morphological dictionary (called a **signature**), a **Minimum Description Length (MDL)** test is conducted. The MDL test is used to accept or reject a given signature as part of the morphology of a given language. The principles applied, the heuristics used in the algorithm and the procedure followed in morphological analysis under Linguistica2001 are all presented in Chapter 2.

However, since Linguistica2001 could not handle morphological analysis task internal to Amharic stems, another approach based on the principle of **autosegmental phonology** is adopted. This approach represents different phonological elements in different tiers and uses association lines to maintain the relation between elements of the different tiers. Applying this principle required algorithms and data structures are designed and a prototype system, called Amharic Stems Morphological Analyzer (**ASMA),** is developed as part of this study. The prototype system was developed using Visual C++. While description of the principle behind is presented in chapter 2, the algorithms and data structures are discussed in chapter 4.

Description of two separate corpuses prepared is presented in Chapter 4. The first corpus comprises 5,236 words was drawn from a fiction by Be'alu (1970). It was used for the experiment with Linguistica2001. The other, which comprises 326 Amharic word stems, was prepared for the experiment conducted with ASMA. The reasons for having two different

corpuses, the problem encountered during corpus preparation and the action taken to correct them are discussed in chapter 4.

Both Linguistica2001 and ASMA use the respective corpuses as input and produce morphological dictionaries (called **signature)** as their outputs. The signatures (named **stem-signature** in the case of ASMA) show the relationships that exist between different morphemic components. Descriptions of the outputs produced are given in Chapter 5.

However, the study is carried out under a number of constraints. The major problems encountered are **absence of a phonological system** to handle phonologically invoked problems such as assimilation, labialization, and palatalization and **unavailability of corpus data**. The impact of these problems in the performance of the systems is observable from the output produced. Due to the small size corpus used, Linguistica2001 cannot analyze the majority of words in the corpus. Moreover, some errors that should have been corrected by phonological system are observable in the output. To minimize the effect of these two factors on the experimental results, only analyzed words are taken as test data. Further corrective measures taken were discussed in Chapter 5.

The test is carried out by examining the outputs produced – by observing the sorts of results the systems provide, and see what they do well and where they err. The test result shows that Linguistica2001 parses successfully 87% of words of the test data (433 of 500 words). This result corresponds to a precision of 95% and a recall of 90%. ASMA has also identified correctly the morphemic compotes of 241 (or 94%) of 255 stems in the test data. The test was conducted based on evaluation of sample outputs given by two linguists. The testing criteria and procedures employed in the study are presented in Chapter 5.

Although the accuracy of the two systems is somewhat acceptable, it can not be conclusive in many respects. The size of corpuses used in the experiment is too small compared with similar research conducted in Spanish, English, French, German, Latin and Italian.

Moreover, there are a number of activities remained or not considered in this study. This study focuses on the **segmentation aspect of morphological analysis**. To make the study complete, further test in Linguistica2001 should be done – using its triage algorithms, to improve the quality of morphological analysis, and the paradigmatic analysis, to identify word classes (or syntactic categories of words) based on signature use patterns. Furthermore, the two separate systems, Linguistica2001 and ASMA, need to be integrated and tested with a large size corpus.

Apart from these limitations, the study has indicated the possibility of employing unsupervised learning approach in the development of Amharic morphological analysis systems. The autosegmental stem analyzer, ASMA, is also an original work developed by the researcher as part of this study. The 5,326 word size corpus prepared in this study could also be used for related researches.

6.2 Recommendations

A number of additional tasks need to be done to develop fully functional morphological analysis system for Amharic. The previous research by Abiyot (2000) along with the current one could serve as a spring board for future works: - Abiyot (2000) tested the rule-based approach and this one the probabilistic, language-independent approach. To make the works complete; further work could be done along the following lines: -

1. Phonological and morphological processes are intermingled in Amharic word formation and conjugations. The current work considered only morphological process. Thus, further study is required in development of automatic phonological system for Amharic. From literature review, it is found that work in **Amharic lexical phonology** essential to handle such processes as assimilations, labialization and palatalization which are common in Amharic phonology.
2. Availability of linguistically well-formed corpus facilitates greatly corpus-based works in a number of linguistic studies (such as phonology, morphology, syntax and semantics). Thus, linguistically proved large-size Amharic corpus need to be maintained and kept open to let researchers use it for their research purpose.
3. This study focused only on the word segmentation part of the task. Further study is thus required to undertake a full-fledged test with Linguistica2001 employing its triage algorithm for fine-tuning the result and the paradigmatic test to identify word categories based on the type of suffixes they use. Such further study is also required to integrate the two separate system used in this study; and to test the integrated system using large size corpuses.
4. A number of works in the literature reviewed reports that morphological system developed using the rule-based two-level morphology approach are successful. Some morphological systems developed for Semitic languages such as Arabic are also reported successful (see Bleesley, 2001). Thus further study is recommended to test the two-level morphology for Amharic morphological analysis and generation.
5. From literature review, it is also found that Inductive Logic Programming (IPA) is proved to be a feasible way to learn linguistic knowledge in many natural language processing activities

including moprhological analysis. Further study is these recommended to test the applicability of this approach in in development of mophological analysis system for Amharic.

Reference

Abiyot Bayou (2000) *Developing Automatic Word Parser for Amharic Verbs and Their Derivation,* M.Sc. thesis, Addis Ababa University, Addis Ababa.

Albright, Adam and Hayes, B. (1999) *An Automated Learner for Phonology and Morphology.* Available at http:// www.linguistics.ucla.edu/people/ hayes/learning/learner.pdf

Allen, James (1995) *Natural Language Understanding*. 2nd ed. California: The Benjamin/Cummings.

Anderson, S.R. (1988) "Morphology as a Parsing Problem," *Linguistics* (26): 521 – 544

Antworth, E.L. (1991) Introduction to Two-Level Phonology Linguistics. Available at http://www.sil.org/pckimmo/two_level_phon.html

Antworth, Evan L. (1994) Morphological Parsing with a Unification-based word grammar: a paper presented at North Texas Natural Language Processing Workshop. *Available at* http://www.sil.org/pckimmo/ntn1p94.html

Antworth, L. Evan. (1990) *PC-KIMMO: A Two-Level Processor for Morphological Analysis*. Dallas, Texas: Summer Institute of Linguistics.

Baye Yemam (1994) *yäamarñña säwasäw*. Addis Ababa: EMPDE

Baye Yemam (1999) "Root Reduction and Extensions in Amharic" *Ethiopian Journal of Language and Literature* (9): 56 - 88

Be'alu Girma (1970) *käadmas bašägär.* Addis Ababa

Beesley, K.R. (2000) Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plan in 2001. Available at: *http://citesser.nj.nec.com/125348.html*

Bender, M. Lionel and Hailu Fulass (1978) *Amharic Verb Morphology*. East Lansing, Michigan: Michigan State University.

Birru Dori (1992) *Customizing CDSIS/IS for Amharic Texts*. Master, M.Sc. thesis, Addis Ababa University, Addis Ababa

Chanod, J.P. & Tapanainen, P. (1995) Tagging French-Comparing a statistical and a constraint based method. In Proceeding of the 7th Conference of the European Chapter of the Association for Computational Linguistics. Also available at: *http://www.citeseer.nj.nec.com/chanod95tagging.html, Internet*

Crystal, David (1991) *A dictionary of Linguistics & Phonetics*. 3rd ed. Cambridge, Massachusetts: Blackwell Reference

Dereje Teferi (1999) OCR *of type Written Amharic text*, M.Sc. thesis, Addis Ababa University, Addis Ababa

Doszkocs, Tamas E (1986) "Natural Language Processing in Information Retrieval," *Journal of American Society of Information Science* 34(4): 191-196.

Ermias Abebe (1998) *Application of OCR Techniques for Recognition of Formatted Amharic Texts*, M.Sc. thesis, Addis Ababa University, Addis Ababa

Ethiopia: Central Statistical Authority (ECSA). 1998. *The 1994 Population and Housing Census of Ethiopia: Results at Country Level.* Vol. 1 Statistical Report48. Addis Ababa

Getahun Amare (1997) *zämänawi yäamarñña säwasw bäqälal aqäraräb*. Addis Ababa

Goldsmith, John (2001a) "Unsupervised Learning of the Morphology of a Natural Language.," *Computational Linguistics* 27(2): 153 – 198 also available at http://humanities.uchicago.edu/faculty/goldsmith.Lingustica2000/Paper/Paper.html

Goldsmith, John (2001b) Probabilistic Models of Grammar: Phonology as Information Minimization. Available at http://humanities.uchicago.edu/faculty/goldsmith/Chiba/Lecture1.html

Goldsmith, John (2001c) Lecture2; Probabilistic Model Grammar: Morphology from a Machine Learning Perspectives. Available at http://humanities.uchicago.edu/faculty/goldsmith/

Grishman, Ralph. (1984) "Natural Language Processing," *Journal of American Society of Information Science* 35 (.5): 291-296.

Habte Mariam, Marcos (1994) "Towards the Identification of the Morphemic Components the Conjugational forms of Amharic," In Proceeding of the 11th International Conference of Ethiopia. Vol. 1

Hudson, G (1999) *Essential Introductory Linguistics*. Oxford, Blackwell

Hundson, G (1986) "Arabic root and Pattern Morphology without tiers," *Journal of Linguistics,* 22: 85 – 122

Karlsson, Fred and Karttunen, Lauri (2001) Sub-Sentential Processing. Available at http://citesser.nj.nec.com/125348.html

Karttunen, L (1994) Constructing Lexical Transducers. In the Proceedings of 15th International Conference of Computational Linguistics. COLING-94 pp. 406 – 411

Katamba, F (1993) *Morphology*. London: Macmillan

Kazakov, Dimater & Manandhar, Suresh (2000) Unsupervised Learning for Word Segmentation Rules with Genetic Algorithms and Inductive Logic Programming. Available at http://citeseer.nj.nec.com/kuzakov00unsupervised.html

Kiraz, G.A. (1994) Multi-Tape Two-Level Morphology: a case study in Semitic non-linear morphology. In COLING-94: Paper presented to the 15th International Conference on Computational Linguistics (1): 180 – 186

Kiraz, G.A. (1995) Computational Analysis of Arabic Morphology. Available at http://citesser.nj.nec.com/125348.html

[48] The latest official statistics available

Koskenniemi, Kimmo (1983) Two-Level Morphology: A general Computational Model for Word-form recognition and production. *Publication 11, University of Helsinki, Department of General Linguistics, Helsinki*

Kurttunen, Lauri. 1983. "KIMMO: A General Morphological Processor." *Texas Linguistic* v. 22: 265-286.

Leslau, W. (1995) *A Reference Grammar of Amharic*. Wiesbaden: Harrassowitz

Mao, Yonghong (1997) "Natural Language Processing Module (Part of Speech Tagging and Sentence Parsing) Laboratory Manual" http://www.csic.cornell.edu/201/natural_language/, Internet.

McCarthy, J (1981) "A prosodic theory of nonconcatenative Morphology," *Linguistic Inquiry,* 12(3): 373 – 418

Mesfin Getachew.(2001) *Automatic Part of Speech Tagging for Amharic Language: An Experiment Using Stochastic Hidden Markov (HMM) Approach*, M.Sc thesis, Addis Ababa University, Addis Ababa.

Million Meshesha (2000) *A generalized approach to OCR of Amharic texts,* M.Sc. thesis, Addis Ababa University, Addis Ababa

Mullen, D (1986) *Issues in the Morphology and Phonology of Amharic: the lexical generation of Pronominal Clitics*, PhD Theses, University of Ottawa, Ottawa.

Nagamatsu, Kenji and Tanaka (1999) *A Stochastic Morphological Analysis for Japanese employing n-Gram and k-NN method.* Available at http://citesser.nj.nec.com/125348.html

Nagamatsu, Kenji and Tanaka, Hidehiko (1999) "A Stochastic Morphological Analysis for Japanese employing Character n-Gram and k-NN methods, " Available at: http://citeseer.nj.nec.com/125348.html, Internet.

Nega Alemayehu (1999) *Development of a Stemming Algorithm for Amharic Texts Retrieval,* PhD Thesis, university of Sheffield, England.

Podolsky, B (1977) "Morphology of Amharic Verb," In *Proceeding of the 5th International Conference of Ethiopia*, 91 – 96

Robertson, A. and Willett, P (1998) "Application of N-grams in Textual Information Systems," *Journal of Documentation* 54(1):48 – 96

Saba Amsalu Tessera. (2001) *The Application of Information Retrieval Techniques to Amharic Document s on the Web*. M.Sc.Theses, Addis Ababa University. Addis Ababa

Salton, G & M.J. McGill (1983) Introduction to Modern Information Retrieval. McGraw-Hill: New York

Trost, Harald (2000) Computational Morphology. Available at http://www.univie.ac.at/~harald/handbook.html

Vit'anyi, P and Li, Ming (1997) on Prediction by Data Compression. Available at http://citesser.nj.nec.com/18812.html

Vit'anyi, P. and Li, Ming (1999) Minimum Description Length, Induction, Bayesiansim, and Kolmogorov Complexity http://www.cw..nl/~paul/papers/ideal.ps

Walther, M (2000) Finite-State Reduplication on One-Level Prosodic Morphology. Available at http://www.uni-marburg.del/linguistik/mal

Warner, J. Amy (1987) "Natural Language Processing," *Annual Review of Information Science and Technology* Vol. 22: 79-108.

Waxwell, M (1994) "Parsing Using Linearly Ordered Phonological Rules," In *Computational Phonology: 1st Meeting of the ACL Special Interest Group in Computational Phonology, Proceedings of the Workshop, pp. 59 – 70*

Wedekind, Klaus (1996) "Which form Predicate all others best? Variations on the Amharic Verb 'theme'". *Ethiopian Journal of Languages and Linguistics (6): 65 - 91*

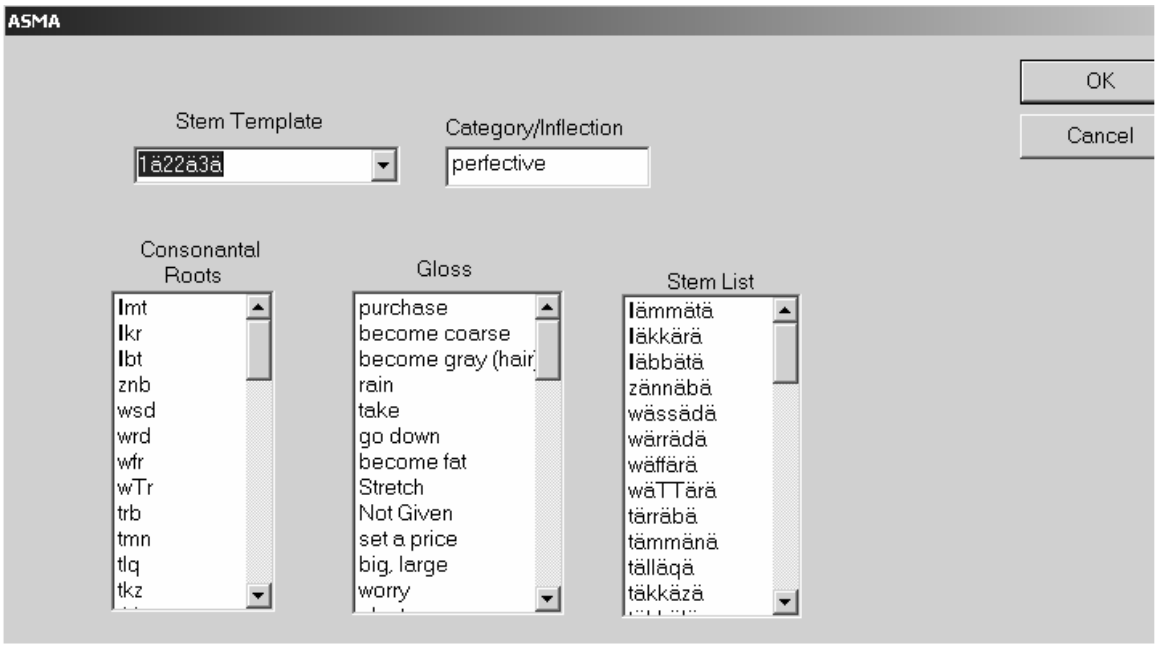
Williams, Paul (1981) On the notion of "Lexical Related' and "Head of a word" *Linguistic Inquiry 12*(2): 245 – 274

Worku Alemu. (1997) *The application of OCR techniques to the Amharic texts, M.Sc t*hesis, Addis Ababa University, Addis Ababa.

Appendix

Appendix 1: Character representation used in the Transcription

Amharic Letter	CONSONANTS																	
	ሀ	ለ	ሙ	ስ	ር	ሽ	ቀ	ብ	ት	ች	ን	ኝ	ክ	ው	ዝ	ዥ	ይ	ድ
Normal	h	l	m	s	r	š	k’	b	t	č	n	ň	k	w	z	ž	y	d
Text	h	l	m	s	r	š	k'	b	t	c	n	n	k	w	z	ž	y	d
Used in the Corpus	**h**	**l**	**m**	**s**	**r**	**š**	**q**	**b**	**t**	**c**	**n**	**N**	**k**	**w**	**z**	**ž**	**y**	**d**
Amharic Letter	CONSONANTS									VOWELS							ዏ	
	ጅ	ግ	ጥ	ጭ	ጵ	ፅ	ፍ	ፕ	ቭ	ኧ	ኡ	ኢ	ኣ	ኤ	እ	ኦ		
Normal	⌘	g	t’	ĉ	p’	s'	f	p		ä	u	i	a	e	□	o	ቀ	
Text	?	g	t'	c	p'	s'	f	p		ä	u	i	a	e	o	o	w	
Used in the Corpus	**j**	**g**	**T**	**C**	**P**	**S**	**f**	**p**	**v**	**ä**	**u**	**i**	**a**	**e**	**ï**	**o**	**u**	



Appendix 2: A Screen Snapshot of ASMA

Appendix 3: A Corpus used for Expeiment with Linguistica2001

Appendix 4: A Corpus used in the experiment with ASMA

Appendix 5: Partial Visual C++ Source Code of ASMA

Since the size of this thesis exceeds the maximum size Scool of Graduate Studies, AAU allows, Appendix 3 - 5 are compiled separately and made availavle at the Bibliographic Laboratory of School of Information Studies, Addis Ababa University.