
ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

# INCORPORATING LINGUISTIC FEATURES IN BI-DIRECTIONAL AMHARIC - ENGLISH STATISTICAL MACHINE TRANSLATION

*By*
TSEGAYE ANDARGIE

FEBRUARY,2019
ADDIS ABABA, ETHIOPIA


ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

---

# INCORPORATING LINGUISTIC FEATURES IN BI-DIRECTIONAL AMHARIC - ENGLISH STATISTICAL MACHINE TRANSLATION

---

*By*
TSEGAYE ANDARGIE

FEBRUARY,2019
ADDIS ABABA, ETHIOPIA


ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

---

# INCORPORATING LINGUISTIC FEATURES IN BI-DIRECTIONAL AMHARIC - ENGLISH STATISTICAL MACHINE TRANSLATION

---

*A Thesis Submitted to School of Information Science in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Science*

*By:*
TSEGAYE ANDARGIE

*Advisor:*
Martha Yifiru(PhD)

February,2019
Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

---

# Incorporating Linguistic features in bi-directional Amharic - English Statistical Machine Translation

---

*By:*

Tsegaye Andargie

Name and signature of Members of the Examining Board

| | | | |
|---|---|---|---|
| Martha Yifru (PhD) | | Oct, 8, 2019 | |
| Advisor | Signature | Date | |
| Wondwossen Mulugeta (PhD) | | Oct 8, 2019 | |
| Examiner | Signature | Date | |
| Million Meshesha (PhD) | million | Oct 08, 2019 | |
| Examiner | Signature | Date | |

# Declaration of Authorship

This thesis has not previously been accepted for any degree and is not being concurrently submitted in candidature for any degree in any other university. I declare that the thesis is a result of my own investigation, except where otherwise stated. I have undertaken the study independently with the guidance and support of my research advisor. Other sources are acknowledged by citations giving explicit references. A list of references is appended.

Signature: _____

Tsegaye Andargie

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

Advisor's Signature: _____

Martha Yifru (PhD)

# *Acknowledgements*

I would like to thank my advisor Dr. Martha Yifru for her constant support, guidance, and patience. It was a great honor to work with her for the last two years. I also would like to thank Prof. Michael Gasser for developing morphology analysis tools for under-resourced languages in Ethiopia, in which this study won't have been possible otherwise.

Throughout these years, I would like to thank my father, Andargie Mekonnen, who have taught me a lack of dedication is disrespect for those who believe in me. Thank you for always motivating and supporting me. I would also like to thank all my fellow teachers at the School of Information Science for sharing your invaluable experiences.

Lastly, my deepest thanks to my family for their moral support and for being always there when needed, without their assistance, I would have not been able to finish.

*Dedicated to late grandpa.* **ርዕሰ-ደብር መኮንን መንግስቱ**

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| BLEU | **Bilingual** Evaluation Understudy |
| BP | Brevity Penalty |
| CBSMT | Corpus-based Statistical Machine Translation |
| EBSMT | EExample-based Statistical Machine Translation |
| EASMT | English Amharic Statistical Machine Translation |
| FSMT | Factored Statistical Machine Translation |
| iOS | internetwork Operating System |
| IRSTLM | IRST Language Modeling |
| LM | Language Model |
| LTS | Long Term Support |
| MT | Machine Translation |
| NLTK | Natural Language Tool Kit |
| NLP | Natural of Language Processing |
| PBSMT | Phrase-based Statistical Machine Translation |
| POS | Part of Speech Tag |
| RBMT | Rule Based Machine Translation |
| SMT | Statistical Machine Translation |
| SL | Source Language |
| TL | Target Language |
| WIC | Walta Information Center |

# Chapter 1

# Introduction

Natural Language Processing (NLP) concerns about giving computers the ability to process human language consisting of speech and language processing, human language technology, computational linguistics, speech recognition, and synthesis and machine translations. From these efforts of making a machine-usable in the arena of human language processing, machine translation has been around for almost six decades by now [1]. In this chapter, we have discussed machine translation, the gap in corpus-based Amharic - English machine translation and a proposed solution to improve the translation accuracy of translating to/from both Amharic and English along with significance and scope of the study.

## 1.1 Background

Machine Translation (MT) is a subfield of NLP that investigate the use of computers to automate some or all the process of translating from one language to another [2]. There are two approaches to machine translation, namely Knowledge (rule) based and Data-driven (machine learning based) approaches [3]. The former has fallen to reach much attention since human languages are rich and complex that it could never be fully analyzed and distilled into a set of rules. The growing nature of every language also makes the latter approach more preferable, that's why it got a much momentum in the research community worldwide which allow the computer to discover the rules by itself along with the translation from a parallel data statistically [4].

In the data-driven approach, early efforts were based to find a sentence similar to the input sentence in a parallel corpus, and make the appropriate changes to its stored translation which was referred as Example-based machine translation. Later statistical machine translation has come around on solid mathematical foundations.

A statistical approach to machine translation is the dominant approach in the field at the moment. In Statistical Machine Translation(SMT) systems are trained on large quantities of parallel data (from which the systems learn how to translate small segments), as well as even larger quantities of monolingual data (from which the systems learn what the target language should look like) [5]. Parallel data is a collection of sentences corpus in two different languages, the source, and target, which is a sentence-aligned, in that each sentence in one language is matched with its corresponding translation in the other languages.

Current trends in SMT have extended the state-of-the-art approach to statistical machine translation, so-called phrase-based models by explicit use of linguistic information which may be morphological, syntactic or semantic [4]. Phrase-based models are limited to the mapping of small text chunks from parallel data. Integration of additional linguistic markup has shown better translation performance, both in terms of automatic scores, as well as in producing grammatical coherence for many language

pairs [6]. This kind of translation models that are developed by incorporating additional linguistic information is called Factored Translation Models.

Experimental results demonstrated significant improvement of translation quality in terms of accuracy and fluency have been reported for language pairs including English to Czech [1], English to Tamil, English to German, English to Spanish and English to Chinese [2], [4].

## 1.2 Statement of the Problem

Amharic is the most spoken and the working language of the federal government in Ethiopia[7].It's a member of the Semitic language family, which has a rich inflectional and derivational morphologies. This means that a single word in Amharic can consist of a lemma and many morphemes each of which represents a different meaning. On the other English is one of the simplest languages in the world with limited inflections[8].

Ordinary Phrase-based statistical methods of translation consider each word form as a separate token in itself independent of other forms of its own. This means that the translation model treats, say, for example, the word "ቤቶች" completely independent of the word "ቤት". Any number of "ቤት" in the training corpus does not add any knowledge to the translation of "ቤቶች" though one is the plural form of the other in the real world. This approach has limited success rates on translating between languages to/from Amharic and English due to a very different syntax and morphology [6]. A promising segmentation of morphemes at a word level has been tried between Amharic and English, though it did not deliver the promised performances for translating to/from Amharic [5]. Amharic also lacks language resources for Natural language processing tasks. So for such naturally unrelated languages, a translation system has to handle both natural differences and data sparsity of the language to improve translation accuracy with the resource at hand efficiently.

There have been some efforts made within SMT to tackle data sparsity for translating from/to Amharic. In a preliminary experiment conducted to translate from English to Amharic using the SMT approach on a domain-specific parliamentary corpus, a 0.34% improvement has been reported by applying morpheme segmentation [5]. The rule-based segmenter used for the study was able to segment a word into smaller morphemes. For example, the Amharic word "በተፈጥሮአዊ" is segmented into three morphemes "በ", "ተፈጥሮ" and "አዊ" each correspond to a preposition, noun, and adjective marker respectively. The segmenter was able to segment prefixes and suffixes. The following words "ሀላፊነትና, ሀላፊነትን, በሀላፊነት, በሀላፊነትና, ከሀላፊነት, ከሀላፊነትና" were all segmented to "ሀላፊነት" by removing the prefix, suffix or both. This Affix segmentation has helped to decrease the number of vocabularies in Amharic corpus by 22% [5].

Statistical machine translation effort made for the target of exploring a genuine approach using limited corpus shows that structural reordering can improve translation accuracy for hierarchical translation. Though structural reordering has helped to improve translation accuracy against non-reordered hierarchical translation, both of these translations experiments has been reported to have less accuracy compared to an ordinary baseline phrase-based translation [6]. Experiments on the same data set, reordering in the phrase-based model have found to be low in performance which has less translation accuracy score than the baseline. Further recommendations by the same authors show that improvements could be made by "applying linguistic features using morphological analyzer and applying syntactic information using language parser for Amharic and English languages" [6].

Relatively recent thesis work in Addis Ababa University department of computer science on bidirectional Amharic to English SMT using constrained corpus was conducted on two different parallel texts, one consisting of simple sentences and the other, complex sentences. These experiments were carried out separately and the result obtained for simple sentences is not far from complex sentences in both Amharic to English and English to Amharic translations [9]. This implies that unless a corpus is very large or Amharic surface forms changed to its base forms(lemma) and linguistic information's incorporated much improvement could not be achieved towards state-of-art translation performances like other languages.

These experiments demonstrate that the currently dominant phrase-based statistical approach cannot solve short-comings that the nature of Amharic language imposes to the field since the problem is not how simple or complex a sentence could be it is rather on how simple or complex a token is in representing a meaning about the other language pair. This imposes that unless each word is explained with the possible natural features, morphologically richness of Amharic will always drain translation accuracy per any language pair whether the target language has a reach or simple morphological inflections.

On the other hand, a highly related study conducted on Tigrigna which, the researcher claims factored translation models tends to be far insufficient for English to Tigrigna factored machine translation, a 16.5% decrement from the baseline system [10]. Contrary to the English-Tigrigna combination, studies in Turkish [11], Arabic [12] [13], Czech [1], and Tamil [2] languages have suggested integrating different linguistic information has a directly proportional effect to increment in translation accuracy. This study reports both morphologically rich and agglutinative languages has merited from linguistic feature incorporation in which all of them have used at-least POS tag and lemma of a word as well as an extra specific feature as to the study. Reports in translation accuracy due to linguistic feature integration and factored translation models has been very promising and assure.

All the above studies and scenarios designated that there is a gap to investigate on how to incorporate proper linguistic features for further improvements it promised to other international under-resourced and morphologically rich languages. The aim of this study is, therefore to investigate how linguistic information can be incorporated to improve the state of art baseline phrase-based translation of English texts into Amharic texts or vice-versa. Each linguistic information might not have the same contribution in the bidirectional translation, that's what it makes a necessity to explore the individual contribution of each to find an effective combination to improve translation accuracy.

## 1.3 Research Questions

Basic questions this research thesis have answered are:

- Does incorporating linguistic features to statistical machine translation models exceed an ordinary phrase-based translation model in translation performance for Amharic and English language pair?
- Which linguistic information has mattered the most from POS tags, Lemma of a word, and Morpheme segments for better translation performance to Amharic and/or English?
- What combinations of linguistic information has better translation performance?

## 1.4 Objective of the study

### 1.4.1 General Objective

The general objective of this study is to integrate linguistic features to phrase-based statistical translation models bidirectionally in Amharic - English translation using Factored translation models.

### 1.4.2 Specific Objectives

The specific objectives of this study to achieve the overall general objective are:

- Collection and Pre-processing of parallel (bi-text) and part of speech(POS) corpus
- Develop different POS tagging models using different approaches and select the best tagging model for Amharic
- Identify and apply the best POS tag model for English
- Identify and adopt the best available word lemmatizer, morphological segmenter, and analyzer for both Amharic(i.e. HornMorpho) and English (i.e. WordNet, Spacy Neural POS Tagger)
- Review and identify techniques that enable to employ linguistic features on parallel corpus bidirectional.
- Train baseline, tagged and segmented translation models by incorporating linguistic features using factored translation models.
- Evaluate, Compare and Contrast the performance of the models to see the impact of a linguistic feature

## 1.5 Significance of the study

### Significance to researchers

The main contributions of this research work is an academic knowledge on how to integrate linguistic features to the state of art baseline phrase-based statistical machine translation and what level of improvement it could add to bidirectional Amharic - English machine translation. The results can be further adapted with the necessary tweaks from other local Semitic languages.

### Significance to other NLP Applications

Further studies on NLP tasks can take this study as an additional component for studies in a speech to text, text to speech and speech to speech translation regarding English - Amharic language pair as well as other Ethio-Semitic languages.

Integrating linguistic features can also help to solve data sparsity problem so that a better translation can be modeled with available insufficient data as compared to another baseline system, on which it needs more data to be on the same level of accuracy. This makes final translation models portable and simple on which handheld devices can also further adopt the factored model for translating electronic reading materials from Amharic to English or English to Amharic to offer cheap, consistent, convenient and fast machine translation for anyone in need.

In a general manner, this study can be integrated to other NLP applications so that individuals, enterprises, Language Service Providers, Law firms and Governments can use it in our society for dissimilation more documents for more audiences. Corporates can also use this with human assistance to aid translation for increase increased throughput. within applications such as in information extraction, document retrieval, intelligence analysis, electronic mail, and much more [14].

## 1.6 Scope and limitation of the study

The scope of this study is limited to Amharic English Statistical Machine Translation using baseline phrase-based models and factored models. The study is conducted bidirectionally. i.e., from Amharic to English and from English to Amharic since each linguistic feature incorporated has a different effect on the vice verse translation. Translation accuracy to a very inflected language and from a very inflected language, i.e. Amharic has been investigated to increase the performance and accuracy in both directions.

All combination of linguistic information has been used to generate factored models separately and a comparison will be made to identify the most linguistic information which matters the most. Linguistic features will be adopted for both the language model and Translation model in the Factored representation. Since one is interdependent over the other.

Due to limited computing resources translation model optimization or model tuning has been set to a maximum iteration of 10 for all systems because tuning is the slowest step of the process. Even in this scenario for factored and tagged feature combinations tunning has taken more days to complete. Check more on our Experimental setup in Chapter 2 for more.

## 1.7 Thesis organization

This thesis is organized into six chapters, the first chapter discuss the introduction, statement of the problem, the objective of the study, scope and limitation of the study, the methodology followed by research design, data collection, an approach for the study and evaluation procedure.

The second chapter deals with a literature review which focuses on the approach of machine translation, machine translation tools and related works previously done locally and abroad prior to this study. Previous studies have been summarized with each other to see what leads and recommendations they could have in common for this study.

The third chapter deals with an overview of the Amharic language and its relationship with the English language and discussion challenges the language has by its nature. The orthography of

the writing system and word formations and morphological characters of Amharic have been discussed.

Chapter four discuss designing processes of the prototype (translation model) including, corpus preparation, types of the corpus used for the study, used POS taggers, morphological analysis, morphological segmentation, how linguistically enriched data is prepared and early preprocessing experiments used to the system.

Chapter five deals with the experiments of the study which include different experiments and the results of the experiments with the interpretation of findings. Which experiments lead to the other and which experiments have little or no effect on the translation accuracy improvement we have been hoping for.

The last chapter covers summaries and conclusion of the findings in this study while articulating recommendations for further studies in the area.

# Chapter 2

# Literature Review

Machine translation (MT), a sub-field under Natural language processing (NLP), is the use of computers to automate all or part of the process of translating from one human language to another natural language [15]. Translating from one language to the other requires a deep and rich understanding of the source language and the text given and sophisticated vocabulary, rule and syntax of the target. Using computers to do so make the problem more difficult, fascinating endeavor to be creative in and tackle.

Although the 1966 ALPAC report conducted on realities of machine translation claims that post-editing machine translation output was not cheaper or faster than full human translation among other things usability of lousy boost the field of study. Despite reduced research efforts due to this report first commercial translations systems came into being in 1976 known as Météo for translating weather forecasts, which was developed at the University of Montreal and is still operating ever since [16].

## 2.1 Why Machine Translation?

Efforts on MT research are not limited to a fully automatic, high-quality translation rather a rough translation is sufficient enough for browsing foreign material. Recent trends are also to build limited MT applications in combination with speech recognition, especially for hand-held devices like Google Assistance and Amazon Alexa.

Major authors in the field of MT categorize its use's broadly into three categories:

(a) assimilation - the translation of foreign material for the purpose of gisting and understanding the content;

(b) dissemination - translating text for publication in other languages for international audience; and

(c) communication - such as the translation of emails, live chat discussions [16] [17] [3].

Machine translation may serve as a basis for post-editing, although efficiency is highly dependent on the quality of MT system post-editor does not need to know the foreign input language. Monolingual speaks are easy to find than bilingual speakers which reduce the cost for correcting the output [16]. In countries like Ethiopia where more than one major languages spoken, the socio-political importance is a favor to count on and get the most out of it [3].

## 2.2 Measures for Selecting Machine Translation Tools

In ordinary statistical phrase-based MT system, a toolkit is evaluated based on how much accurate it is to a reference human translation and how much time and resource does the data structure took to achieve it are the two main measures to evaluate the performance effectiveness and efficiency [16]. However, in our case of expressing tokens linguistically, the linguistic quality and ease of integration with the existing tools are the indicators to favor one over the other. Linguistic quality is examined by how much the target output it is consumable by higher-level applications concerning the source intention or means that translated output can take less time to post-edit with the translation management system. Moses [18] on this regard comes on top as it is the most widely adopted state of art machine translation toolkit in the area of SMT.

## 2.3 Machine Translation Approaches

Early researchers focused on building translation systems using hand-written linguistic rules which were created by expert linguists. That approach is called rule-based machine translation. Due to the burden of handcrafting rules with so many decisions those which are hard to formalize, a new data-driven example-based translation system that has been built especially in Japan, 1980 trying to find a sentence similar to the input sentence in a parallel text, and make the appropriate changes to its stored translation [16].

After the late 1980s, a new approach arose for the machine translation problem in the labs of IBM Research. Researchers started to develop translation systems using parallel texts of language pairs and use statistics. Since then, research on statistical machine translation has rapidly grown [17].

### 2.3.1 Rule Based Machine Translation (RBMT)

Rule-based machine translation (RBMT) is a knowledge-based MT which retrieves rules from bilingual dictionaries and grammars based on linguistic information about the source and target languages. RBMT generates target sentences based on syntactic, morphological and semantic regularities of each language in which the linguistic rules are built on. It converts source language structures to target language structures and it is extensible and maintainable as in [15] [19].

There are three methodologies of RBMT systems (see Figure 2.1), namely:

1. Direct Approach
2. Transfer Approach and
3. Interlingua Approach

#### 2.3.1.1 Direct Approach

This is the oldest approach in RBMT in which source language words are translated without passing through an additional/intermediary representation. The words will be translated as a

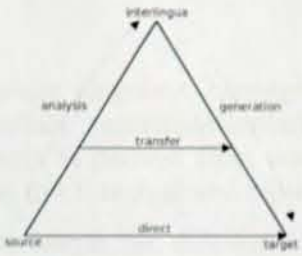


FIGURE 2.1: Methods of Rule based Machine Translation (Source - Learning MT [19])

normal manual dictionary does word by word, usually without much correlation of meaning between them. Dictionaries and grammars will be used to analyze the source language as well as to synthesize the target-language text. Meaning SL⇒TL transformation is the function of dictionary lexicons and language syntax.

Language divergence is a common problem in Direct Translation approach on which, lexically and syntactically similar sentences of the source language are not translated into sentences that are similar in lexical and syntactic structure in the target language.

E.g. English source text –"just got arrived now" can be translated to "አሁን ደረሰ" or "ደረሰ" in Amharic on which the phrase converges to a single word.

#### 2.3.1.2 Transfer Approach

In Transfer based systems, morphological and syntactical analysis is done on the source language and syntactic/semantic structure of source language is transferred into the syntactic/semantic structure of the target language. Here source language text is converted into less language specific representation and the same level of abstraction is generated with the help of grammar rules and bilingual dictionaries. In the transfer approach of translation divergence, there is a transfer rule for transforming a source language (SL) sentence into a target language (TL), by performing lexical and structural manipulations. (see Figure 2.2)

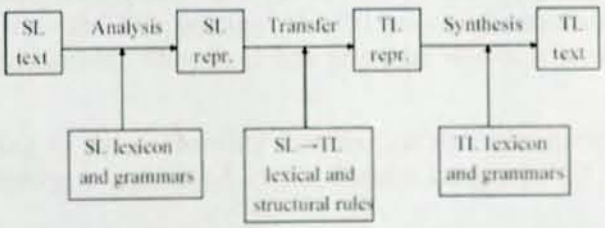


FIGURE 2.2: Transfer based RBMT Adopted from adopted from Jaiswal & Ballabh, a study on MT methods [20]

The transformation process starts with morphological analysis on which surface forms of the SL input text are classified as of their part of speech tag or sub category (gender, number, tense). Then lexical transfer is done using basic dictionary translation the source language lemma, while finally the TL surface forms are generated using morphological generators of the target language.

While Transfer based approach seems to be more promising than direct translations, the number of rules will grow drastically in case of general non-domain specific translation systems.

#### 2.3.1.3 Interlingua approach

In Interlingua approach, it tries to make linguistic homogeneity across the world. Source language is translated into an intermediary representation which does not depends on target or any other language. A target language is derived from this assisting form of representation assuming that SL text concept exists in the intermediary representation.

Although emphasizes on single representation for different languages, the main challenge in Interlingua approach is that the definition of an Interlingua is difficult and maybe even get impossible for a wider domain.

### 2.3.2 Corpus Based Machine Translation (CBMT)

To overcome the problem of knowledge acquisition of rule based machine translation and need of highly skilled linguists, a new alternative approach for machine translation emerges at IBM lab in the 1980s by using already available translation using solid mathematical foundations by modeling translation task as a statistical optimization problem [16]. Corpus Based Machine Translation (CBMT) uses, a bilingual parallel corpus to obtain knowledge for new incoming translation. This approach uses a large amount of raw data in the form of parallel text. This raw data contains source text and their translations. These corpora are used for acquiring translation knowledge [21]. Corpus based approach is further classified into the following two sub approaches Statistical Machine Translation and Example-based Machine Translation Approach.

#### 2.3.2.1 Statistical Machine Translation Approach

Statistical machine translation (SMT) is generated on the basis of statistical models, based on Bayes Theorem, initially takes the view that every sentence in target language is a possible translation of any sentence in the source and the most appropriate is the translation that is assigned the highest probability from the parallel text by the system [16] [21]. The idea is to find the most probable translation of a given sentence. This approach can be applied to any language combination that has enough parallel text and requires the least amount of human effort among all approaches. Though it has previous routes it was reintroduced by IBM researchers in 1988 [22].

A text is translated according to the probability distribution function indicated by p(f|e), which is the Probability of translating a sentence f in the Source Language (SL) $f$ to a sentence e in the Target language(TL) $e$.

The problem of modeling the probability distribution p(e|f) has been approached in a number of ways. One intuitive approach is to apply Bayes theorem (2.1).

$$p(e|f) = \frac{p(f|e)p(e)}{p(f)} \tag{2.1}$$

That is, if p(f|e) and p(e) indicate translation model and language model, respectively, then the probability distribution p(e|f) $\propto$ p(f|e)p(e). The translation model p(f|e) is the probability that the source sentence is the translation of the target sentence or the way sentences in E get converted to sentences in $F$. The language model p(e) is the probability of seeing that TL

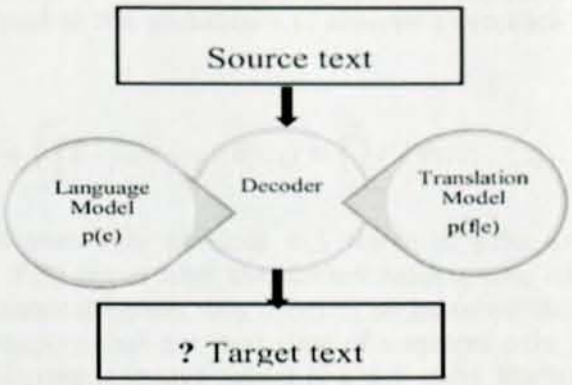


FIGURE 2.3: How SMT works?

string or the kind of sentences that are likely in the language $E$ [23]. This decomposition is attractive as it splits the problem into two sub problems. Finding the best translation ê is done by picking up the one that gives the highest probability as shown in Equation (2.2):

$$\hat{e} = \arg\max_{e \in e^*} p(e|f) = \arg\max_{e \in e^*} p(f|e)p(e) \tag{2.2}$$

From the formula (2.2), we can see that statistical machine translation problem actually has three parts, as explained in the mathematics of MT [19]:

1. building a target language model to estimate $p(e)$;
2. building a translation model to estimate $p(f|e)$;
3. searching for a translation e to maximize the product $p(f|e)p(e)$, which is also called decoding [16] [21].

For a rigorous implementation of this one would have to perform an exhaustive search by going through all strings $\hat{e}*$ in the native language. Performing the search efficiently is the work of a machine translation decoder that uses the foreign string, heuristics and other methods to limit the search space and at the same time keeping the acceptable quality. Thus SMT depends on a language model, a translation model, and a decoding algorithm. The translation model ensures that the machine translation system produces the target hypothesis corresponding to the source sentence. The language model ensures the grammatically correct output [21] which all can be described by the following architecture. (see Figure 2.3)

Alongside with source language text for translation, the language model which ensure that words come in the right order, the translation model which assigns a probability that a given source language sentence will likely be translation target language sentence are used by the decoder to compute the one target sentence with maximum probability. Here the language models are used for assigning a probability to the occurrence of a sequence of m linguistic units (mostly words or phrases), by means of the probability distribution of all units. The model, where the probability of a unit depends on the previous n units, is called an N-gram language model.

In N-gram language model, the probablity of a sentece S with words $W_1$, $W_2$,$W_3$ ...$W_m$ is shown in Equation (2.3) defined as the probability to observe a sentence in an n-gram language model [24].

$$P(w_1,\ldots,w_m) = \prod_{i=1}^{m} P(w_i|w_1,\ldots,w_{i-1}) \approx \prod_{i=1}^{m} P\left(w_i|w_{i-(n-1)},\ldots,w_{i-1}\right) \tag{2.3}$$

An N-gram language model uses only previous n-1 words of prior context to estimate the probability of a given word. This comes from the Markov Assumption, which is the presumption that the future state of a dynamical system only depends on its recent history[15]. In particular, a $k^{th}$-order Markov Model suggests that the next state of a system only depends on the k most recent states, therefore an N-gram language model is a $n$-1 order Markov model. That means, the probability of observing a word $w_i$ in a sentence where previous $i$-1 words are known, can be approximated to the probability of observing it in the context of previous n-1 words. However, when an unseen word is confronted, this model will fail and assign a probability of 0 to the new word. To eliminate this problem of 0 probablity, smoothing methods are usually applied, such as Kneser-Ney smoothing [24] which is also used in this study.

In any translation system after the language model is computed, the translation model is created using the bi-lingual parallel corpus. The first step in creating the translation model is the word alignment. After the words are aligned, two major statistics are derived from alignments; fertility and distortion [23]. Fertility is the number of target language words generated for a source language word. Distortion is the position difference between the target language word and the source language word in the sentence.

Searching is done after finding all possible translations of a given sentence. Using language and translation models created above, probabilities for partial alignments are computed. Basically its a process of stacking the promising partial alignments, which have higher probabilities and extend the stack until a complete translation is achieved for a given sentence $S$.

In SMT, Phrase-based approach is the most common statistical method in use[16]. It's as a joint probability method for learning words and phrases from the bilingual corpora. In the classic phrase-based approach, a word is represented as a single token where as later variants of it allow other linguistic information to go with it. These are syntax-based [25], hierarchical [26] and factored [4] phrase-based approaches [27].

#### 2.3.2.1.1 Factored phrase-based approach

In factored phrase-based approach, a word is represented as a vector of factors each of which serves as different levels of annotation. Figure 2.4 is adopted from [4] to give an illustration of factored representation of words.

In this study, for a bidirectional English - Amharic translation system, we adopt the factored phrase-based approach and explored its potentials for these language pairs.

As phrase-based models, factored translation models can be seen as the combination of several components (language model, reordering model, translation steps, generation steps). These components define one or more feature functions that are combined in a log-linear model:

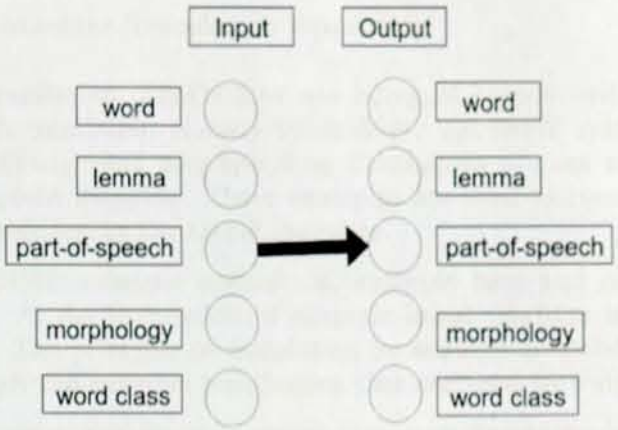


FIGURE 2.4: Representations of input and output in factored SMT(source - Koehn & Hoang [4])

$$p(\mathbf{e}|\mathbf{f}) = \frac{1}{Z} \exp \sum_{i=1}^{n} \lambda_i h_i(\mathbf{e}, \mathbf{f}) \tag{2.4}$$

Since Z is a normalization constant in Equation (2.4) it is ignored in practice. To compute the probability of a translation $e$ given an input sentence $f$, we have to evaluate each feature function $h_i$. For instance, the feature function for a bigram language model component is (m is the number of words ei in the sentence e):

$$\begin{aligned} h_{\mathrm{LM}}(\mathbf{e}, \mathbf{f}) &= p_{\mathrm{LM}}(\mathbf{e}) \\ &= p\left(e_1\right) p\left(e_2|e_1\right) .. p\left(e_m|e_{m-1}\right) \end{aligned} \tag{2.5}$$

If we we specifically see the feature functions introduced by the translation and generation steps of factored translation models. The translation of the input sentence f into the output sentence e breaks down to a set of phrase translations $\left\{\left(\bar{f}_j, \bar{e}_j\right)\right\}$.

For a translation step component, each feature function hT is defined over the phrase pairs $\left\{\left(\bar{f}_j, \bar{e}_j\right)\right\}$ given a scoring function [16]:

$$h(\mathbf{e}, \mathbf{f}) = \sum_j \tau\left(\bar{f}_j, \bar{e}_j\right) \tag{2.6}$$

For a generation step component, each feature function hG given a scoring function $\gamma$ is defined over the output words $(e_k)$ only:

$$h_{\mathrm{G}}(\mathbf{e}, \mathbf{f}) = \sum_k \gamma\left(e_k\right) \tag{2.7}$$

The feature functions follow from the scoring functions( $\tau, \gamma$) acquired during the training of translation and generation tables [4].

#### 2.3.2.2 Example-based Machine Translation Approach

Example-based machine translation (EBMT) also use bilingual corpus with parallel texts as its main knowledge, in which translation is done by analogy. An EBMT system is given a set of sentences in the source language and corresponding translations of each sentence in the target language with a point to point mapping. These examples are used to translate similar types of sentences of the source language to the target language.

There are four tasks in EBMT: example acquisition, example base and management, example application and synthesis. At the foundation of example-based machine translation is the idea of translation by analogy. The principle of translation by analogy is encoded to example-based machine translation through the example translations that are used to train such a system [21].

| English | Amharic |
|---|---|
| How much is that red shoe? | ያ ቀይ ጫማ ምን ያህል ነው? |
| How much is that small camera? | ያ ትንሽ ካሜራ ምን ያህል ነው? |
| How much does that car weigh? | ያ መኪና ክብደቱ ስንት ነው? |
| How much price for that shoe? | ያ ጫማ ዋጋው ስንት ነው? |

TABLE 2.1: Sample EMBT by of a minimal pairing Amharic vs English phrases (source-Martha Yifru [28])

The example in the above table above shows an example of a minimal pair, meaning that the sentences vary by just one element. These sentences make it simple to learn translations of sub sentence units. EBMT is an attractive could yield better translation but it requires analysis and generation modules to produce the dependency trees needed for the examples database plus computational efficiency is another bottleneck especially for large databases [21].

#### 2.3.2.3 Hybrid Machine Translation Approach

By taking the strength of both statistical and rule-based translation methodologies, one can use both together, namely a hybrid-based approach, which has proven to have better efficiency in the area of MT systems [21].

The hybrid approach can be used in several different ways. In some cases, translations are performed in the first stage using a rule-based approach followed by adjusting or correcting the output using SMT information. In the other way, rules are used to pre-process the input data as well as post-process the statistical output of a statistical-based translation system. This technique is better than the previous and has more power, flexibility, and control in translation and proven serious efficiency several governmental and private based MT sectors [21].

## 2.4 Related Studies

English, being currently the dominating language for communicating internationally, it is essential to prioritize the language pairs involving English as the source or the target language in translation tasks. With this view, research on machine translation on the Amharic-English (or English-Amharic) language pair has increased in recent years.

However, statistical machine translation from English to Amharic is still challenging in many aspects. Because of the linguistic differences between these languages discussed in Chapter 3 Section 3.5, building a robust machine translation system for this language pair is harder than doing it for linguistically closer languages. Baseline Machine Translation track shows that systems built upon linguistically close language pairs are more successful than the other systems in terms of BLEU scores. On the hand, linguistically different languages need further processing than baselines to get the same or near results [13].

There are a considerable amount of studies, both published and unpublished, done to solve the problem of Machine translation for Amharic with other languages. These researches were conducted in different languages employing a variety of different approach and methodology for both local and foreign languages.

In this study, we have reviewed key studies done locally and globally specifically on machine translation to or from Amharic or vice versa to any language pair, along with sibling Ethio-Semitic, Semitic and other globally related morphologically rich languages.

### 2.4.1 Studies on Ethiopian languages

Here listed are major researches conducted on Amharic and other similar Ethio-Semitic languages.

#### Preliminary Experiments on English-Amharic SMT (EASMT)

This experiment was conducted by Mulu and Besacier [5] in Addis Ababa University IT Doctoral study, following the statistical approach which relies heavily on bilingual parallel aligned corpora of the source and target languages. Following the baseline phrase-based models of SMT, they collected the English-Amharic parallel corpus from parliamentary documents that exist online plus those collected manually are used for the preliminary experiment on EASMT.

Preprocessing tasks have been conducted on each corpus found from Federal Negarit Gazeta to retain and convert the full content into a valid Unicode text format suitable for the MOSES SMT system. Further alignment tasks were conducted using Hunalign along with manual trimming tasks.

The experiment has been conducted using 18,432 English-Amharic sentence pairs with a total of 500K English and Amharic tokens extracted from the parliamentary corpora to measure the accuracy of the translation system. Accordingly, the baseline phrase-based BLEU score result was 35.32%. Further by applying morpheme segmentation to the Amharic result set a 0.34 score increase in BLEU has been achieved which is 0.92% increase compared to baseline [5].

#### Making Amharic to English Language Translator for iOS

Because there was still no language translator for Amharic-English language pair on the iOS platform, Hana developed an SMT model which was later consumed by Microsoft Translator Hub and accessed using Microsoft Translator APIs so that it is more accessible from phone.

She adopted 100k freely available Muslim Quran corpus from OPUS free content by selecting sentence whose token length is between 8- 18 resulting in a BLEU score 7% due to fewer quality data. As a software engineer practitioner, the author claims it was enough to proceed to develop

iOS bundles and HCI modules for Amharic characters. To improve the poor translation quality the author concludes to improve quality of the parallel text.

### English to Amharic Machine Translation Using SMT

Like the other similar studies, this paper also deals with the translation of English to Amharic using statistical methods. Ambaye and Yared [6] have conducted the research aiming to be a standpoint for similar researches. They have conducted the study by applying both phrase-based and hierarchical approaches of SMT a first attempt to study both approaches using the same data.

After prepossessing of a raw data collected from Ethiopian constitution Negarit Gazeta Awaj, Bible books and some international regulations and Ethiopian governmental portals for a final of 37,970 bilingual sentences were collected. Additionally 68, 815 monolingual sentences were also collected from known web news agency like Ethiozena, Ethiopian reporter, Walta information center and Addis Admas using web mining for language modeling [6].

Using Moses, Giza++ and IRSTLM as a major toolkit a translation accuracy in terms of BLEU Score of 18.74 was achieved through Phrase-based translation models while the same data hierarchical translation model system scores with BLEU of 8.43 even if better in reordering than phrase based. The researchers further select only simple English phrases to see their output on translation accuracy. To do so they select 12,537 simple English phrases which are translated as Amharic words the score is improved to 23.16 for the phrase-based model and 11.24 for hierarchical model [6].

### Bidirectional English-Amharic MT: An Experiment using Constrained Corpus

This research work also implements a statistical machine translation approach. In order to realize the goal, two different corpora were prepared and collected; the first corpus consisted of simple sentences and the other, complex sentences. Two language models were developed, one for Amharic and the other for English so as to ensure a bi-directional translation.

Simple sentences corpus was made manually and the other complex sentence corpus was taken from the Bible and Public Procurement directive of the Ministry of Finance and Economic Development of Ethiopia. The former consists of around 1,020 simple sentences while the other consist of 3,488 complex sentences. Corpus was verified by a certified linguist for proper categorization and correctness, which makes it by far the first of its kind to use linguistically checked corpus.

The experiments were taken separately, one for the simple sentences and the other for complex sentences. The result obtained for the simple sentence using BLEU Score had an average of 82.22% accuracy for the English to Amharic, 90.59% for the Amharic to English. For the complex sentences, the result acquired from the BLEU Score was approximately 73.38% for the English to Amharic, 84.12% for the Amharic to English [9]. As can be noted from the results this study has the maximum score of all studies reported due to the nature of the data. Simple constrained corpora tend to work in favor of the translation accuracy in this study since these data are manufactured.

### Bidirectional Tigrigna – English Statistical Machine Translation

This study is by far the most recent work conducted on Ethio-Semitic language specifically on Tigrigna – English SMT explored improving translation accuracy by applying linguistic information [29].

Experiments were conducted in three sets baseline phrase-based machine translation system, morph-based employing unsupervised morphology leaning and post-processed segmented systems. The researcher collected digitally available Tigrigna and English versions of some chapters of the New Testament of Holy Bible and FDRE constitutions texts. Using these 6,000 parallel texts collected, dividing 90% for training and the reaming 10% for testing, he has found a translation accuracy of 18.65% for English – Tigrigna and 36.40 Tigrigna – English using baseline phrase-based machine translation. Using the same data set the unsupervised morpheme-based MT accuracy has degraded to 13.44% and 35.66% respectively.

Considering that the segmentation learner used is unsupervised, the segmentation is based on the frequency of the morphs within a limited corpus, the author decided to post-process the segmentation model so that it only focused on segmenting Tigrigna conjunctions and prepositions namely 'ን', 'ብ', 'ከ', 'ን ን', 'ውን', referring to "For", 'By', 'While', 'And' and 'Also' in the English. The result obtained from the post-processed experiment has outperformed the other two experiments, by a BLEU score of 22.46 % for English – Tigrigna and 53.35 % for Tigrigna – English and translations [29] suggesting that applying preprocessing and post-processing techniques help for translation accuracy for Ethio-Semitic languages.

### English -Tigrigna Factored Statistical Machine Translation

But the author also claims the low performance of the factored system is due to the POS tagger and morphology analyzer used in the study. POS tagger used was trained using 1,018 manually tagged words prepared by the researcher himself without any linguist advisory which might not be greater than 100 sentences compared to 17,649 parallel texts. The stemmer used also performs very low which left most of the corpus unsegmented propagating the error to the translation model, not even checked with gold standards rather only the researcher corpus. Despite all the deficiency of resources a 16.5% translation accuracy of factored models was a great accomplishment but international works promise an improvement using factored models as discussed in the literature below.

A factored English to Tigrigna translation was conducted using a Statistical machine translation approach using 17,649 sentences collected from VOA news and the Bible. This data has a total of 500,000 tokens used for training and testing translation system. The author conducted his experiment employing three types of corpus namely baseline, Segmented and factored corpus that integrates linguistic knowledge at the word level. Preprocessing, morphological segmentation, stemming and POS tagging were performed to prepare the factored corpora based on the researcher judgment.

The factored corpus has shown a decrease 4.53% from the baseline system on which the baseline phrase-based translation has a translation accuracy of 21.04% while the factored has 16.5%. The researcher believes that the low performance of the factored system is accounted to the POS tags attached since the tagger was trained using a small 1,018 words manually tagged and prepared by the researcher without any linguist advisory which might not be greater than 100 sentences compared to 17,649 parallel texts which have 500k tokens. The unpublished stemmer

used also performs only 85.8% which left 15% of the corpus unsegmented propagating the error to the translation model, not even checked with gold standards rather only the researcher corpus [10].

Constituting all these problems it is by far to the researcher knowledge, the only paper to report a decrease to translation accuracy by integrating linguistic information to the compared to the baseline system.

#### What makes previous local studies similar?

The major data sources for researches conducted so far on Ethio-Semitic languages, for parallel corpora are the Holy Bible, FDRE Constitution, FDRE Criminal code, the regional state constitution, and international conventions. All of the researchers claims data collection was challenging and spent more time on the collection and preprocessing more than on the actual experiments. If there was a standardized corpus, which can be used as a baseline for conducting and evaluating studies from one another it could have been easier the compare and contrast.

It is hard to compare results each study have reported due to the nature of the difference in the data adopted for each study but we can see all the results are not satisfactory when compare to state of art SMT results. In a matter of fact that incorporating linguistic features has no been done for Amharic is a huge endeavor to explore for the seeking of knowledge due to the nature of the language and data sparsity between Amharic and English.

### 2.4.2 Studies on Foreign languages

Here listed are major researches conducted on Semitic languages abroad and other Morphological Rich Languages.

#### Linguistic Factors in Statistical Machine Translation Involving Arabic Language

Arabic has a rich morphology compared to the English language and is considered as one of the morphologically rich languages. This fact adversely affects the performance of English-Arabic SMT. Phrase-based SMT models have a limitation of mapping phrases or blocks from the source to the target languages without any use of linguistic information [12]. In this study, the author incorporating linguistic tools, specifically, the use of POS tagging incorporated as a linguistic feature in a factored translation model and its impact on translation quality for English-Arabic machine translation is reported.

Experiments were carried out on the Arabic English Parallel News Text Part 1corpus available free in Linguistic Data Consortium (LDC), contains 68,685 news text sentence pairs from the Linguistic Data Consortium catalog, 2 Million Arabic words and 2.5 Million English words aligned at the sentence level. The English and Arabic corpus were tagged with the Stanford Log-linear POS Tagger using the Penn Treebank tag set [12].

Results were compared on translation quality obtained from the baseline system, which contains only the surface form of the words, with the morphologically extended system by the POS model. For the baseline system, a 60.95% score for BLEU and POS model revealed scores of 63.94% on which system with POS factor improved the translation quality with 2.99% BLEU scores over the standard surface-based system.

### Factored Statistical Machine Translation System for English to Tamil Language

Highly rich morphological nature of the Tamil language makes automatic machine translation to English a challenging task. Morphologically rich languages need extensive morphological preprocessing before the SMT training to make the source language structurally similar to target language [30].

Since English and Tamil languages have disparate morphological and syntactical structure, the authors' proposed work is to develop a machine translation system which pre-processes the English language sentences according to the Tamil language. These pre-processed sentences are given to the factored SMT models for training and the output converted back to its surface word from using Tamil morphological generator.

Experiments were conducted with nine different types of models, which are trained, tuned and tested with the help of general domain corpora of 6,500 parallel sentences for training, 1462 for testing and 500 used for tuning along with developed linguistic tools. The baseline phrase-based model using the surface forms of the words without any additional linguistic knowledge which has a BLEU score of 1.07% was outperformed by Factored System + Rule base Reordering + Compounding + Morph Generator which has BLEU score of 4.14%. The factored model has even exceeded the online Google Translate by 6.66% measured by a Multi-BLEU scoring.

### English-to-Czech Factored Machine Translation

Being a Slavic language with very rich morphology Czech has free word order. To handle rich morphology of the language the author has used a News Commentary corpus of 55,676 pairs of sentences for exploring the effect of factored translation model in translation accuracy for English-to-Czech machine translation [1].

The Czech part of the corpus was tagged and lemmatized using local proprietary Czech toolkit while the English part was tagged using MXPOST and lemmatized using the Morpha tool. This study incorporates lemma and morphology of after basic pre-processing along with the baseline input word forms to output word form translation. Two types of language models have also been used, a 3-gram LM over word forms and a 7-gram LM over morphological tags. This experimental setups using Moses toolkit reduce the risk of early pruning, the generation step operationally precedes the morphology mapping step.

All linguistic factors incorporations experimented with multi-factored phrase-based translation aimed at improving morphological coherence in English-Czech MT output has show result improvement in BLEU scores by explicit modeling of morphology and using a separate morphological language model to ensure the coherence. Using multiple linguistic features together a score of 12.9 has been improved to 14.9 by incorporating full Czech POS tags, lemmas, and morphology, a 21% improvement over the baseline [1].

### Head Finalization and morphological analysis in factored phrased-based SMT from English to Turkish

In this study, an approach for translating from English to Turkish is introduced by incorporating linguistic features. Turkish is an agglutinative language with a free constituent order, whereas English is not agglutinative and the constituent order is strict plus there have been reported for lack of data between these two languages [31].

This study represents English and Turkish at morpheme-level but also applying a Head Finalization reordering technique which was successfully used for other languages, which are grammatically similar to Turkish. The author has divided the experiment into six for comparison, baseline, POS tagged and factored with a combination of the token lemma and morphology segments and then reordered of this there experiments. The reordering was conducted before adding any linguistic information so as to match English and Turkish word orders.

Using 54,391 sentences pairs from the European Union, European Court of Human Rights documents and several treaty texts the author reported improvements in reordering and factored morpheme segments representation, an increased BLEU score from a baseline score of 19.62 to 30.93, which corresponds to an increase of 57% [31].

## 2.5 summary

While there are complications on the studies conducted including Ethio-Semitic languages as a source or a target, factored studies on Semitic languages like Arabic have paid off significant improvements over baseline systems. Similarly, studies on Tamil and other morphologically complex and agglutinative languages like Turkish and Czech have reported major improvements.

On the contrary, the factored model for Tigrigna - English translation did not bring any improvement to the performance of the SMT system. As in the study, these are because of the underdeveloped POS taggers and Tigrigna stammer used [10]. A POS tagger developed on 1,018 words and stemmer with less vocabulary coverage, has less prediction to a general document in real-world. The low accuracy has propagated to the factored models and they have negatively affected the translation accuracy by misleading the translation tables in training.

While both Amharic and Tigrigna are under-resourced, Amharic has better NLP resourced compared to Tigrigna. We are going to compare and contrast the best methodologies to adopt than in the Tigrigna to avoid pre-underdeveloped assumptions for our translation. There are also POS tagging and lemmatization studies to adopt for Amharic than Tigrigna. Considering all the above we expect significant improvement for Amharic-English language pairs when incorporating linguistic features due to the nature of the language and the available resources.

# Chapter 3

# Amharic Language

## 3.1 Overview of Amharic Language

Amharic (አማርኛ Amareña) is a Semitic language that is spoken mainly in Ethiopia. Though many dialects are spoken throughout Ethiopia (including Amharic, Tigrinya, Oromiffa/Affan Oromo, etc), Amharic is the most popular and widely used. Since it is the working language of the Ethiopian government, it has gained an official status and it is used throughout the country [7] [32].

There are more than ninety languages that are spoken in Ethiopia (according to the 1994 Ethiopian census conducted by Ethnologue). Amharic is spoken by more than 17 million people, which is about one-third of Ethiopia's population (and another third speak the Oromo language). It has been the language of the court and the dominant population group in Highland Ethiopia since the late 13th century. It is spoken to some extent in every province, including the Amhara region [7].

The history of Amharic language traces back to the 1st millennium B.C. to the days of King Solomon and the Queen of Sheba. Historians explain that immigrants from southwestern Arabia crossed the Red Sea into present-day Eritrea and mixed with the Cushitic population. This union resulted in the birth of Ge'ez (ግዕዝ), which is the language of the Axum Empire of Northern Ethiopia. It existed between the 1st Century A.D. and the 6th Century A.D. When the power base of Ethiopia shifted from Axum to Amhara between the 10th Century A.D. and the 12th Century A.D., the use of the Amharic language spread its influence, hence becoming the national language.

Amharic is also one of the most widely studied languages in Ethiopia. It is also used as a medium of instruction for primary level education in Addis Ababa, the capital city of Ethiopia. It is part of the school curriculum in most elementary and secondary levels of education. It is also studied in various universities in America and other developed countries as an elective course.

Amharic belongs to the Semitic group (like Arabic and Hebrew) within the Afro-Asiatic family of languages. About 50 other Semitic and non-Semitic Afro-Asiatic languages are spoken in Ethiopia alongside Amharic. Unlike Arabic, Hebrew or Syrian, Amharic is written from left to right.

Like other Semitic languages, Amharic has a very elaborate verb morphology. An Amharic verb root consists of a set of (usually three) consonants. Depending on the tense, and other grammatical features, the consonants may be separated by particular vowels and possibly geminated (doubled). A verb form normally also has one or more suffixes and possibly one or

more prefixes as well, agreeing with the subject and sometimes the direct or indirect object of the verb. Complicating things further (at least for the adult second-language learner), there are at least ten different classes of verbs, each modifying its stem in a different way for the different forms. Like Japanese and many other languages, Amharic is a verb-final language. Amharic nouns are relatively simple by comparison, though they may take suffixes indicating possession ('my', 'his', etc.), plural, and a few other grammatical functions.

Unlike most African languages, Amharic has been a written language for many years, at least 500. It is written using a syllabic writing system that is unique to Ethiopian Semitic languages. Compared to other African languages, Amharic has a fairly sizable written literature [33].

Since Amharic is the most widely studied language [34], [35] compared to other local languages, we describe only the Amharic verb Morphology and sentence structure which is the most relevant to our study.

## 3.2 Amharic Orthography

Amharic is written in Ethiopic or Fidel(ፊደል), which is the writing system also used by Tigrigna. Unlike Arabic, Hebrew and Syriac, which have their vowel signs written independently above, below, or within the letters, the Ethiopic writing system attaches its vowel signs to the body of the consonant, so that there are as many modifications of the form of each consonant as there are vowels. Amharic is a syllabary writing system where each character represents an open CV syllable, i.e., a combination of a consonant followed by a vowel [36] [33].

The Ethiopic alphabet has 33 basic characters. Each such character is modified in some regular fashion to reflect the seven vowels of the language. Therefore, there are in total 33 x 7 = 231 characters. Even though the Amharic alphabet is Unicode standard, it is sometimes convenient to represent it in ASCII. Written in SERA [36] (System for Ethiopic Representation in ASCII), the basic characters which can also be called the consonants of the language are in alphabetic order:

```
C = [h, l, H, m, s, r, 's, x, q, b, t, c, 'h, n, N, a, k, K, w, 'a, z, Z, y, d, j, g, T,
C, P, S, 'S, f, p]
```

As one can see in the Appendix A the vowels are

```
V = [e, u, I, a, E, I, o]
```

In some cases, more than one constant can be used to represent a sound in Amharic. So Amharic identifies 28 unique sounds out of 33 basic constants. Those repeated constants are:

- [h, H, ʻh ]= [ሀ, ሐ, ኀ ]
- [s, ʻs ]= [ሠ, ሰ ]
- [S, ʻS ]= [ጸ, ፀ ]
- [a, ʻa ]= [አ, ዐ ]

The letters in each set represent the same sound. For example, an Amharic word that has the letter ʻhʼ can be written in three equivalent ways and it is important to recognize this letters in NLP tasks, they must be treated as one [37]. For a list of all letters see Appendix A.

When it comes to punctuation marks, some in Amharic are as similar in English. But there are major differences; e.g. in Amharic, four points (2 consecutive colons) are used to mark the end of a sentence plus English uses comma while Amharic uses a colon with a bar. Amharic follows the SOV language pattern agreement where words are separated by space. Except in poems, the head verb is usually at the end of a sentence [37].

## 3.3 Amharic Morphology

Amharic has a complex morphology. Word formation involves pre-fixation, suffixation, in-fixation, reduplication, and Semitic stem inter-digitation. Like other Semitic languages, Amharic verbs and their derivations constitute a significant part of the lexicon. In Semitic languages, words, especially verbs, are best viewed as consisting of discontinuous morphemes that are combined in a nonconcatenative manner. Verbs are commonly analyzed as consisting of root consonants, template patterns, and vowel patterns is a major character of Semitic languages. Except for very few verb forms (such as the imperative), all derived verb forms take affixes to appear as independent words [38].

Most function words in Amharic, such as Conjunction, Preposition, Article, Relative marker, Pronominal affixes, Negation markers, are bound morphemes, which are attached to content words, resulting in complex Amharic words composed of several morphemes. Nouns inflect for the morph syntactic features number, gender, definiteness, and case. Amharic adjectives share some morphological properties with nouns, such as definiteness, case, and number. As compared to nouns and verbs, there are fewer primary adjectives. Most adjectives are derived from nouns or verbs. Amharic has very few lexical adverbs. Adverbial meaning is usually expressed morphologically on the verb or through prepositional phrases. While prepositions are mostly bound morphemes, postpositions are typically independent words [38].

Amharic verbal stems, consist of a 'root + vowels + template' merger. For instance, the root verb sbr + ee + CVCVC leads to form the stem seber ('broke'). In addition to such non-concatenative morphological features, Amharic uses different affixes to create inflectional and derivational morpheme. Affixation can be prefix, infix, suffix, and circumfix. To study the word-formation of Amharic language through its morphological complexity, it's better to understand by looking at the word-formation process through inflection and derivation.

### 3.3.1 Word formation

Amharic morphology is complex, particularly verbs employing not only prefixes and suffixes but also modifications of the typical consonantal root-and-pattern type [39]. Amharic noun and adjectives are inflected for case, number, gender, and person whereas verbs are inflected for person, number, gender, tense-aspect-mood (TAM) and case. Verbs may also contain a pronoun object marker. The order of the affixes is fixed in the language in such a way that the subject agreement comes right after the stem, followed by the direct object agreement and then by the dative (benefactive, olfactive, instrument marker) and finally by the indirect object agreement. Any exchange of the position would result in an exchange of role in the grammatical functions of the respective agreement affixes [40].

#### 3.3.1.1 Inflectional behavior

**Verb**

Verbs are inflected for person, gender, number, aspect, tense, and mood [32]. For indicating a person, gender and number suffix are added to the stem, see Table 3.1.

1. Person, Number, Gender, Case

| Person(Subjective Case) | Gender | Singular | Plural |
|---|---|---|---|
| First | | ሰበር-ኩ | ሰበር-ን |
| Second | Masculine | ሰበር-ክ/ህ | ሰበር-ኣችሁ |
| | Feminine | ሰበር-ሽ | ሰበር-ኣችሁ |
| Third | Masculine | ሰበር-ኧ | ሰበር-ኡ |
| | Feminine | ሰበር-ኧች | ሰበር-ኡ |

TABLE 3.1: Inflectional behavior of Amharic verbs

2. Tense / Aspect

   Tense has a different format, present tense in the Amharic language is indicated using the verbs "ነው ". For past tenses "ነበር" is used. For continuous tense form "እየ-" prefix is used. If we insert a verb "ነው " or "ነበር " to the end of Amharic sentence along with continues form indicator, the form will have present continues or past continues tense form respectively.

   Other inflectional behavior, shown in Table 3.2 is indicated by benefactive and olfactive indicators.

| | | Benefactive | Malfactive |
|---|---|---|---|
| 1st person | Singular | ሰበር-ኧልኝ | ሰበር-ኧብኝ |
| | Plural | ሰበር-ልን | ሰበር-ብን |
| 2nd person | Masculine | ሰበር-ልህ | ሰበር-ብህ |
| | Feminine | ሰበር-ልሽ | ሰበር-ብሽ |
| | Plural | ሰበር-ላችሁ | ሰበር-ባችሁ |
| | Polite | ሰበር-ኦሎት | ሰበር-ኦቦት |
| 3rd person | Masculine | ሰበር-ኧለት | ሰበር-ኧበት |
| | Feminine | ሰበር-ኧላት | ሰበር-ኧባት |
| | Plural | ሰበረ-ላችው | ሰበረ-ባችው |
| | Polite | ሰበረ-ላችው | ሰበረ-ባችው |

TABLE 3.2: Inflectional behavior of Amharic verbs (benefactive and malfactive)

3. Mood

   There are four moods in Amharic: declarative, interrogative, negative and imperative. Verbs can take different forms according to the mood as shown in Table 3.3.

**Noun**

Amharic nouns can be marked on numbers, definiteness, gender, and case (Yimam, 2000).

| | | Declarative | Interrogative | Negative | Imperaltive |
|---|---|---|---|---|---|
| 1st person | Singular | ሰበረ | ል-ሰበር | አል-እ-ሰበር | |
| | Plural | -ን-ሰበር | -ን-ሰበር | አል-ን-ሰበር | |
| 2nd person | Masculine | ት-ሰብር | ት-ሰብር | አት-ሰበር | ሰበር |
| | Feminine | ት-ሰበር-ኢ | ት-ሰበር-ኢ | አት-ሰበር-ኢ | ሰበር-ኢ |
| | Plural | ት- ሰብር - ኡ | ት-ሰብር-ኡ | አት-ሰበር--ኡ | ሰበር-ኡ |
| | Polite | ት-ሰብር-ኡ | ት-ሰብር-ኡ | አት-ሰበር--ኡ | ሰበር-ኡ |
| 3rd person | Masculine | ይ-ሰበር | ይ-ሰብር | አይ-ሰብር | ይ-ሰብር |
| | Feminine | ት-ሰበር | ት-ሰብር | አት-ሰብር | ት-ሰብር |
| | Plural | ይ-ሰበር-ኡ | ይ-ሰብር-ኡ | አይ-ሰብር-ኡ | ይ-ሰብር-ኡ |
| | Polite | ይ-ሰበር-ኡ | ይ-ሰብር-ኡ | አይ-ሰብር-ኡ | ይ-ሰብር-ኡ |

TABLE 3.3: Inflectional behavior of Amharic verbs (mood)

1. Number

Noun number markers are "-ኦች" suffix for the noun which ends with a consonant and"-ዎች" suffix is used to the noun with vowel ending. For indicating personal pronoun and the proper noun "እነ-" prefix is used. The plural form can be formed by repetition, for instance, the plural form of "ቅጠል" will be "ቅጠላቅጠል" and the borrowed nouns from Geze is formed by "-ኦች","-አን" or "-አት" morphemes[41]. Sample inflection by number are shown below in Table 3.4.

| Noun in singular Form | Description of the Noun | Morpheme | Plural Form |
|---|---|---|---|
| በግ | ending with consnanat | -ኦች | በግ + -ኦች = [በጎች] |
| ተማሪ | ending with vowel | -ዎች | ተማሪ + -ዎች = [ተማሪዎች] |
| አንተ | personal pronoun | እነ- | እነ- + አንተ = [እናንተ] |
| ቅጠል | plural formation by repetation | | ቅጠል-ኡ-ቅጠል=[ቅጠላቅጠል] |
| መምህር | loan words from Geez | | መምህራን |
| አንበሳ | | | አናብስት |

TABLE 3.4: Amharic Noun Inflection by Number

2. Definiteness

Definiteness indicated by the affixation of morphemes or vowels based on the number, gender, and/or ending of the noun. A morpheme that indicates a singular masculine is"-ኡ" suffix and singular feminine are indicated by using "-ዋ" or "-ኢቱ" suffixes. The plural marker is "-ኦች-ኡ". The above morphemes are used for the nouns with consonant endings. On the other hand, the noun which has vowel ending will have "-ው" suffix for singular masculine indicator and singular feminine "-ዋ" or "-ይቱ" is used. For the plural indicator "-ዎች-ኡ" is used as shown in the examples on Table 3.5.

3. Gender

Gender defnitive is as smilar as the above number defnitive by affiation of the morpheme -ኢት or -ኡ.
e.g.

- በግ - ኢት = [በጊት]
- በግ -ኡ = [በጉ]

| Indefinite Noun | Ending of the Noun | Number | Gender | Definite Noun |
|---|---|---|---|---|
| በግ | Consonanat | Singular | Feminine | በግ -ዋ = [በግዋ] / በግ -ኢቱ = [በጊቱ] |
| | | | Masculine | በግ -ኡ = በጉ |
| | | Plural | | በጎች-ኡ = [በጎቹ] |
| አህያ | Vowel | Singular | Feminine | አህያ -ዋ = [አህያዋ] / አህያ -ይቱ = [አህይይቱ] |
| | | | Masculine | አህያ-ው = [አህያው] |
| | | Plural | | አህያ-ዎች-ኡ = [አህያዎቹ] |

TABLE 3.5: Definiteness by affixation of morphemes

- ላም - ዋ = [ላሚ]
- በሬ -ው = [በሬው]

4. Case

Case indicator can be an objective case by using "-ን" or possessive case by the affixation of morphemes or vowels based on a person, number, gender, and/or ending of the noun (personal pronouns by prefixing "የ-"). see examples listed on Table 3.6 about possessive cases for both singular and plural Amharic Nouns.

| Subjective case | Ending of noun | Person | Number | Gender | Possessive case |
|---|---|---|---|---|---|
| በግ | Ending with consonant | First | Singular | | በግ-ኤ(በጌ) |
| | | | Plural | | በግ-አችን(በጋችን) |
| | | Second | Singular | Masculine | በግ-ህ(በግህ) |
| | | | | Feminine | በግ-ሽ(በግሽ) |
| | | | Plural | | በግ-አችሁ(በጋችሁ) |
| | | Third | Singular | Masculine | በግ-ኡ(በጉ) |
| | | | | Feminine | በግ-ዋ(በግዋ) |
| | | | Plural | | በግ-አችን(በጋቸው) |
| አህያ | Ending with vowel | First | Singular | | አህያ-ዬ (አህያዬ) |
| | | | Plural | | አህያ-አችን (አህያችን) |
| | | Second | Singular | Masculine | አህያ-ህ (አህያህ) |
| | | | | Feminine | አህያ-ሽ (አህያሽ) |
| | | | Plural | | አህያ-አችሁ (አህያችሁ) |
| | | Third | Singular | Masculine | አህያ-ው (አህያው) |
| | | | | Feminine | አህያ-ዋ (አህያዋ) |
| | | | Plural | | አህያ-አቸው (አህያቸው) |

TABLE 3.6: Inflectional behavior of Amharic Nouns

## Adjective

Adjectives are marked by numbers, gender, definiteness, and case (Yimam, 2000). The number marks are "-አች" for consonant ending and "-ዎች" for vowel ending. Repetition of consonant could also mark the plural form of adjectives like "ረዥም" to "ረዥ -ኡ- ዥም"[ረዣዥም]. "-ኢት" is used for gender marker and "-ን" is used to mark the case (objective case). The definiteness marker is marked by using morphemes or vowels based on the number, gender or ending of the adjective. The representations of definiteness marker are specifically to adjectives are discussed by samples on Table 3.7.

| Indefinite Adjectives | Vowel/Consonant | Number | Gender | Definite Adjectives |
|---|---|---|---|---|
| አዲስ | Consonant | Singular | Feminine | አዲስ-ዋ (አዲስዋ) / አዲስ-ኢቱ (አዲሲቱ) |
| | | | Masculine | አዲስ-ኡ (አዲሱ) |
| | | Plural | | አዲስ-አች-ኡ (አዲሶቹ) |
| አሮጌ | Vowel | Singular | Feminine | አሮጌ-ይቱ (አሮጌይቱ) |
| | | | Masculine | አሮጌ-ው (አሮጌው) |
| | | Plural | | አሮጌ-አች-ኡ (አሮጌዎቹ) |

TABLE 3.7: Inflectional behavior of Amharic Adjectives (definiteness marker)

#### 3.3.1.2 Derivational behavior

Verb

Amharic verbal stems (from which various forms of verbs are formed) can be derived from:

1. Verbs can be derived from verbal root by affixing the vowel "ኧ" to produce CVCCVC, for instance, "ስብር" to "ስኧ-ብብኧር" [ስበር] and by repeating penultimate consonants and affixing the vowels" ኧ" and "አ" to produce CVCVCCVC-, e.g "ፍልግ" to "ፈላለግ" by "ፍ -ኧልአልልኧግ" derivation pattern. Verbs can also be derived from the verbal stem by affixing morphemes like "ተ" "አስ" or "አ".
2. Verbal Stems by affixing morphemes

| Verbal Steam | Morpheme | Derived Verbal Stem |
|---|---|---|
| ሰበር— | ተ— | ተ—ሰበር = [ተሰበር—] |
| ለመን— | አስ— | አስ—ለመን = [አስለመን—] |
| ወረድ— | አ— | አ—ወረድ = [አወረድ—] |

TABLE 3.8: Affixing Morphemes in Amharic verb derivation

3. Compound Words of

   Stems and verbs, e.g. ስብር + አለ = ስብር አለ

   Sub-words and verbs, e.g. ፀጥ + አደረገ = ፀጥ አደረገ

In general Amharic verbs show a high degree of inflection since the person, case, gender, number, tense, aspect, mood and others are marked on the verb. For example, አልገደለንም indicates:

- The subject እሱ (third person, masculine, singular)
- The object እኛን (first person, plural)
- Negation አል...ም
- Past tense ገደለ

Noun

Nouns are derived from other nouns, adjectives, roots, stems, and the infinitive form of a verb by affixation and intercalation[41]. The "-ነት", "-ኧኛ", "-ኧት", "-አዊ", "-ተኛ" "-ኛ"

and" ባለ-"affixes are used to derive nouns from other nouns. A noun that is derived from adjective will take "-ኧት" and "-ነት" suffixes. Nouns can also be derived from verbal roots by intercalation and affixation. For instance "ነገር" is derived from "ን-ግ-ር" by "ን-ኧ-ግ-ኧ-ር" pattern of derivation.

Amharic nouns can be derived from:

1. Verbal Roots by infixing vowels between consonants (C) as shown below (see Table 3.9)

| Verbal Root (Examples) | Pattern of Derivation | Derived Noun |
|---|---|---|
| ጥ-ቅ-ም | CኣCኣC | ጥአቅአም = [ጥቅም] |
| ም-ር-ት | CኣCC | ምኣርት = [ምርት] |
| ሀ-ም-ም | CኣCኧC | ሀኣምኧም = [ሀመም] |

TABLE 3.9: Verbal Roots by infixing

2. Adjectives by suffixing bound morphemes(see Table 3.10)

| Adjective (Examples) | Morpheme | Derived Noun |
|---|---|---|
| ደግ | -ነት | ደግ + -ነት =[ደግነት] |
| ቅርብ | -ኧት | ቅርብ + -ኧት = [ቅርበት] |
| ብልሀ | -ኣት | ብልሀ + -ኣት = [ብልሃት] |

TABLE 3.10: Adjectives by suffixing bound morphemes

3. Stems by prefixing or suffixing bound morphemes(see Table 3.11)

| Stem (Examples) | Morpheme | Derived Noun |
|---|---|---|
| ጠቀም | -ኤታ | ጠቀም + -ኤታ= [ጠቀሜታ] |
| ዝርፍ | -ኢያ | ዝርፍ + -ኢያ= [ዝርፊያ] |
| ድርግ | -ኢት | ድርግ + -ኢት= [ድርጊት] |
| ችል | -ኣታ | ብልሀ + -ኣት = [ብልሃት] |

TABLE 3.11: Adjectives by suffixing bound morphemes

4. Stem-like Verbs by suffixing the bound morpheme-ታ(see Table 3.12)

## Adjectives

Adjectives are derived from nouns, stems or verbal roots[41]. The suffixes "-ኣም" "-ኧኛ" "-ኣዊ" "-ኣማ" are used in the derivation of adjectives from nouns. Adjectives also derived by attaching morphemes to the bound stem using "-ኣ" "-ኤ" and "-ኢታ" suffixes. Adjective those are derived from verbal roots by intercalation and affixation like "ደረቅ" is derived from "ድ-ር-ቅ" by "ድ-ኧ-ር-ኧ-ቅ" pattern of derivation. see sample derivation of Adjective from Nouns on Table 3.13.

| Stem (Examples) | Morpheme | Derived Noun |
|---|---|---|
| ዝም | -ታ | ዝም + -ታ = [ዝምታ] |
| ደስ | -ታ | ደስ + -ታ = [ደስታ] |
| ጨዋ | -ታ | ጨዋ + -ታ = [ደስታ] |

TABLE 3.12: Verbs suffixing the bound morpheme-

| Nonn | Morpheme | Adjective |
|---|---|---|
| ተራራ | -አማ | ተራራ + -አማ = ተራራማ |
| ሰላም | -አዊ | ሰላም + -አዊ = ሰላማዊ |
| ውሸት | -አም | ውሸት + -አም = ውሸታም |

TABLE 3.13: derivations of Adjectives from nouns

#### 3.3.1.3 Compound

Compound words can be derived by affixing "ኧ" and "አ" morphemes resulting noun. Classes of compound words that derive nouns are (a) noun + noun, (b) noun + "ኧ" + noun, (c) noun + verbal stem, (d) verbal stem + "አ" + verbal stem or (e) verbal stem +"አ" + noun. Adjectives can be derived from compound words by affixing " ኧ" to noun and adjective for instance "ሆደ - ኧ- ሰፊ". Compound words are formed by (a) stem and verbs [ስብር- አለ] (b) sub-words and verbs [ፀጥ - አደረገ] to give adjective words [42]. e.g.

- ሆድ - ኧ- ሰፊ = [ሆድ-ሰፊ]
- ልብ - ኧ- ሙሉ = [ልበሙሉ]

## 3.4 Amharic Grammar - Syntax

As we discuss Morphology and Word formation the above section 3.3, neither language structure nor translation is done at a word level. Rather a full meaning is conveyed by a sentence which is an aggregate of words expressing judgment of the mind. The constituent parts of every sentence are a subject, a verb, an attribute, and an object. According to Baye [32] sentences can be classified as simple or complex, while Isenberg [42] classified sentences into three categories namely: simple, complex and compound.

Simple sentences are composed of a subject, an object, and a verb. e.g. "ምድር ሰፊ ናት። ". meaning "the world is wide." very simple. On the other hand, complex sentence amplified by qualifying word connection with the subject or the attribute. Compound sentences have either the subject or the attribute or the object or all of them are included by additional or explanation parts.

The most usual word order in Amharic is subject-object-verb (SOV). However, if the object is tropicalized it may precede the subject, as a result, it will have object-subject-verb (OSV) order. SOV order needs to be agreed to write a meaning full sentence [39]. Subject and Verb needs to be agreed on the person, number, and gender. This agreement also works for subject-object agreement. Adjective and noun should also agree on number and gender. The other agreement in the language is an adverb – verb agreement. Usually, adverbs indicate time, therefore adverb

and verb need to agree on tense. Each of the agreement rules in Amharic is discussed in detail below.

## Subject-verb agreement

The subject is the reigning parts of every sentence and in every sentence, the subject precedes the attribute and the verb. Verbs describe the action by adding a remark on to mood, tense and gender [42]. Subject and verb need to be agreed on gender, number, and person.

e.g. [ሳባ መፅሐፎቹን ሸጠች።።] translated to [Saba sold the books.]

Here in the example, the subject is "Saba" which is a third-person singular and the object is "books" which is a noun with a plural marker "-och". finally a verb "sold" has a feature which indicates a third person singular feminine marker as well as past tense. We can see that the subject and the verb are the same in number feature marker which is singular and in-person agreement both indicates the third person. And also both have a feminine in gender agreement. Although English demands the same kind of agreement one can easily see from the translation it has a subject-verb-Object (SVO) word order.

## Object – verb agreement

Object and verb take the same condition as subject-verb agreement. It needs to be agreed on gender, number, and person.

e.g. [ሳባ ብርጭቆዎቹን ሰበረቻቸው።።] translated to [Saba breaks the glasses.]

The above example explains the object and verb agreement in number. The object "glasses" is in the plural form and the verb "breaks" has a plural object marker.

e.g. 2: [ሳባ ለዮናስ መፅሀፎቹን ሰጠችው።።] translated to [Saba gave the books to Jonas.]

This example demonstrates the gender feature agreement of object and verb. The verb ሰጠችው[gave] indicate a subject and object marker. The verb has features that indicate the subject agreement which is a third-person singular feminine marker and third-person singular masculine object marker. The object ለዮሀንስ[for Jonas] is a third-person singular masculine which agrees with the verb-object marker. Therefore the sentence is grammatically correct.

## Adverb - verb agreement

Adverbs are modifiers of verbs, which mostly are hard to pick a translation for, in morphologically rich languages [4]. Adverbs can be classified into time, place and circumstance marker. Time indicates a given action in which it takes place. Verbs also indicate the time at which action takes place in relation to the adverbs. The time adverb and tense disagreement are one of the common Amharic grammatical errors. The correct type of adverb should be used for the verb and vice versa.

e.g. [ካሳ ወደ ቤት ዛሬ ይመጣል።።] translated to [Kasa will come home today.]

In the above example, the adverb "ዛሬ"[today] which is in the future tense and the verb "ይመጣል"[will come] is also in a future tense. Therefore adverb and the verb agree on time

but if we say "ካሳ ወደ ቤት ትላንት ይመጣል።።" ( Kasa will come yesterday). This is grammatically incorrect because the adverb "ትላንት" and"ይመታል" do not agree in time. The adverb indicates a past tense and the verb indicates a future activity.

### Adjective – noun agreement

Adjectives modify nouns by appearing before them. An adjective often agrees with the subject in gender, number, and case. In number agreement, the noun that the adjective modifies can be in plural form but the adjective can either be in singular or plural form. For instance, ደህና መፅሓፍት meaning good books, ደህና is an adjective singular form whereas መፅሓፍት is a noun plural form.

Adjectives most frequently used in the masculine form. The masculine form of adjectives can modify a feminine noun but not vice versa. For example ክፉ ሴት (bad girl), in this phrase "ክፉ" is an adjective with a masculine marker and the noun "ሴት" is in singular feminine form. But we cannot say ንፅህት ወንድ (clean boy) in which "ንፅህት is an adjective in singular feminine form and the noun "ወንድ" is masculine singular form. This shows that there is no title grammar of Amharic language.

### Incorrect word order

Amharic has a subject-object-verb word order but in a certain way, the words can be written in object subject-verb order. OSV can convey semantically meaning but it is not the formal grammar rule [43].

e.g. [ወርቁ ዮሃንስ ክፍሉን እንደሚረብሽ ያምናለል።።] translated to [Worku belives Yohannes will disturb the class.] The POS tag order for this sentence in the example is as follows, NP NP NP V V.

e.g. [ዮሃንስ ክፍሉን እንደሚረብሽ ወርቁ ያምናለል።።], which has O-[NP NP V] S-NP V-POS tag order.

The first example is in an SOV word order. Such a pattern can be obscure the relationship, especially in a more complex sentence with several modifiers. It is perhaps this possibility which motivates the OSV pattern of a sentence like in the second examples.

## 3.5 Challenges

Amharic poses its challenges to natural language processing at all levels of linguistic studies: phonology, morphology, syntax, semantics, and discourse. For machine translation, the challenges result mainly from the nature of the writing system (see Section 3.2), the complexity of the morphology(see Section 3.3), differences in word order and lack of resources.

### 3.5.1 Writing system challenges

Amharic word is represented by hidden vowels and visible consonants with form Changes [33]. This decease knowledge gain about the word for any statistical tools due to hidden letter features. this has not been the case for English though, because English is written as it is pronounced using only 25 letters.

### 3.5.2 Word Order Problem

In English, the constituent order is strict and it has a Subject-Verb-Object order, while Amharic has commonly constituent order of Subject-Object-Verb [32]. Both Amharic and English have strict word order structure.

However, in English, there are a few situations where the order is changed. For example, temporal adverbs can be used at the beginning or at the end of a sentence depending on the emphasis. Word order difference is not only at sentence level but can also be seen in sub-sentential constructions like phrases and clauses.

### 3.5.3 Morphological challenges

Amharic has a rich inflectional and derivational morphology. This means that a single word in Amharic can consist of a lemma and many morphemes each of which represents a different meaning. Besides, the same morpheme can change form in different words depending on vowel order, consonant assimilation or other phonological processes as discussed in above section 3.3. Thus, the Amharic word can be aligned with a bunch of words in English. An example of this is shown in Table 3.14.

| Amharic Word | Amharic Morphological Representation | English Translation |
|---|---|---|
| ያገኘናቸው | የ-{አገኘ}-ን-አቸው | the things we found |
| ሲያጭበረብሩን | ስ-ይ-{አጭበረብር}-ኡ-ን | they were cheating us |
| የምንጨርሳቸው | የም_-እን-{ጨርስ}-አች_ው | things we are going to finish |

TABLE 3.14: An example of Amharic morphology and English translations

The example in Table 3.14 shows that to build a machine translation system, many to one alignment from English words to Amharic word(s) may be required. Morphological analysis and segmentation are performed on Amharic data to aid with this challenge. Figure 3.1 shows a possible alignment between an English phrase and the morphological decomposed representation of an Amharic word.

An Example in Figure 3.1, is another sample which shows how morphological processing could help improve translation form Amharic (source) to English translation as a target.

As shown in Figure 3.1, the direct matches in this sample reference translation are the word "ምክንያት" and punctuation mark "።" translated to "reason" and "." respectively for English. The others need some sort of morphological decomposition processing to get the exact translation as for example:
"ላለመጨረሳቸው" which is a single verb word in Amharic can be translated to multiple words in English due to its morphological nature, decomposed to "ለ-አል-{መጨረስ}-አች_አው"

- "completing" for the verb "መጨረስ"
- negation word "not" from morphme "አል"
- Subject reference "them" from Suffix morphem "አች_ኣው"
- and conjunction "for" from Prefix morphem "ለ"

Source: የብር ማጠር ስራውን ላለመጨረሳቸው ምክያት ሆነባቸው።
Target: For them, lack of money was the reason for not completing the task.

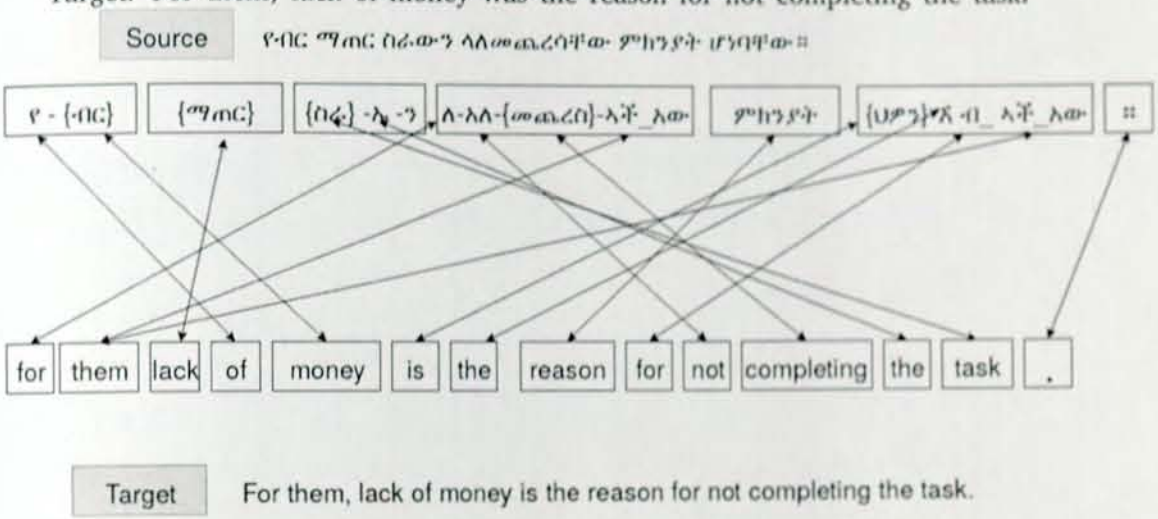


FIGURE 3.1: English words alignment with Amharic morphology

### 3.5.4 Available Parallel Corpora

Compared to other international language pairs, there is only one single corpus for English - Amharic. It is a corpus prepared at Addis Ababa University, under a thematic research project of collecting parallel corpora for bi-lingual English-Ethiopian language pairs available for the research [44]. We have been participating in this corpus collection with our colleagues. The corpus is multi-lingual parallel corpora for English to other five Ethiopian languages including Amharic, Tigrigna, Afan-Oromo, Wolaytta, and Ge'ez.

This corpus has 40,726 English - Amharic parallel texts, which still makes the language very under-resourced. This limited corpus makes it even harder to conduct researches in the area of MT that involve to/from Amharic to/from other languages that have significantly different word orders and morphologies like English as discussed above.

The corpus has been analyzed to see the relationship between English and Amharic sentences based on the categories from which the data is collected [45]. As it can be seen from Table 3.15, there is a significant difference between the morphology of English and the Amharic language. Due to this difference, this study uses linguistic processing specifically, morphological segmentation to tackle the inherent problem due to the difference like Amharic and English languages.

| | History | Legal | Religion | |
|---|---|---|---|---|
| | | | Bible | Blog |
| English | 35,325 | 85,526 | 767,989 | 80,505 |
| Amharic | 29,804 | 63,940 | 472,294 | 62,436 |

TABLE 3.15: Distribution of Amharic and English text in the corpus.

# Chapter 4

# Methodology and Approaches

This study aimed at incorporating linguistic features phrase-based bidirectional Amharic - English statistical machine translation system to improve the performances. Towards that, various data collection and pre-processing tasks have been performed and different tools were selected to be used for the experiment. Sections in this chapter explain the data collection, pre-processing, morphology analysis, POS tagging, factored data preparation and Language Modeling before training the actual target translation model.

### 4.0.1 Research Methodology

Our objective in this study requires training models, integrating and experimenting linguistic feature, recursive evaluation and detailed study towards a knowledge artifact. We have followed a design science [46] sequence process model to model the best linguistically motivated statistical translation model for bidirectional Amharic - English translation. Design science allows us to follow a framework in exploring knowledge personalized to the objective of this study. The design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts which are knowledge's in our case on how to integrate linguistic feature and their role in SMT from Amharic to English or English to Amharic translation. Thus the research method employed in this study is a design since research methodology.

### 4.0.2 Literature review

Secondary data sources, books, journal articles, publications and other previously written resources related to the topic have been referred from researches conducted in the same area. Related works from Addis Ababa university repository has also been reviewed. Limitations, if any, of these studies and the approaches they have used has been thoroughly reviewed.

studies conducted in other related morphologically complex, under-resourced languages and Semitic languages which share the same behavior with Amharic have also been carefully selected and reviewed.

## 4.1 Data Collection Methods

The development of machine translation relies on parallel corpora of the source and target languages plus a monolingual corpus. The monolingual corpus is used to train the language model of the target language, in this case, English and Amharic since the system is bidirectional.

In this study, We have used a portion of a recent corpus prepared at Addis Ababa University, under a thematic research project of collecting parallel corpora for bi-lingual English-Ethiopian pairs available for the research community [44].

The corpora are available for English to other five Ethiopian language pairs, namely Amharic, Tigrigna, Ge'ez, Afan-Oromo, and Wolaytta. Only the English - Amharic segment of the corpora was adopted for this study. This corpus has **40,726** English - Amharic parallel texts which have gone under preprocessing tasks [44]. It is part of the published first attempt data collection for under-resourced languages project that we and our classmates, teachers, and colleagues have been working on in parallel with this study.

Data with an equal number of **30,678** sentences which are from religious biblical data have been used for the development of baseline and other models that incorporate linguistic features so that comparisons could be made based on the same data-set for translation performance and accuracy dependent only on the incorporated linguistic features.

The full Amharic - English part contains a multi-domain History, Legal, Bible and Blog data but due to limited computational resources, we have selected only the biblical part out of the whole English - Amharic corpora. An example from the parallel data is shown in Figure 4.1.

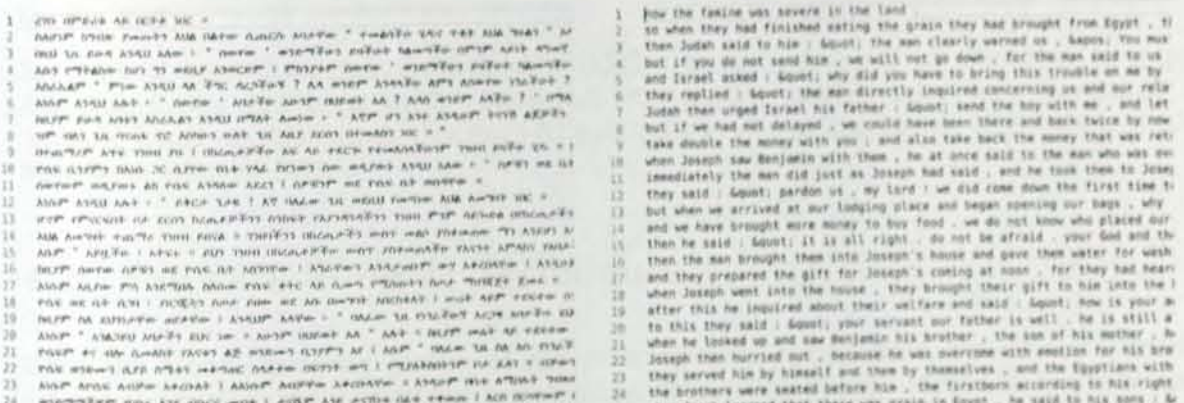

FIGURE 4.1: An illustrative segment from the parallel corpus

Within the parallel corpus, the English corpus contains **767,654** tokens, an average of **25** words per sentence and the Amharic corpus consists of **473,153** tokens, an average of **15** words per sentence. Detail distribution of the parallel data used in this study is shown in Table 4.1. As

it can be noted from the average sentence length in the table, a single Amharic word has a much higher probability to be translated to more than one word in English.

| | Tokens | Word Types | Avg. Sentence Lengeth |
|---|---|---|---|
| Amharic | 473,153 | 73,907 | 15 |
| English | 767,654 | 39,658 | 25 |

TABLE 4.1: Distribution of Amharic and English text.

The corpus has been preprocessed through character normalization to replace a set of redundant characters(ሀ, ሐ, ኀ, ሠ, ሰ, አ, ዐ, ጸ and ፀ) with similar functionality into a single most frequently used character, plus sentences are tokenized and aligned through the merging of verses and removal of unnecessary links, numbers, symbols and foreign texts which are identified as unnecessary by the researchers, where this information have no role in the meaning of the document [44].

For the experiments, the datasets have been divided into 24,511 sentences of training, 3,068 sentences of tuning and 3,067 sentences of testing. The sentences are selected randomly based on 80% - 10% - 10% sampling respectively from the complete corpus before applying any linguistic information.

Using this data on Table 4.1 a total of eight major statistical machine translation systems were created throughout this study. These systems are divided into three categories: baseline, tagged and factored systems. Each category consists of two systems, one built to translate to Amharic and the other to English from their counterparts. Table 4.1 gives a summary of the data used for the study.

For part of speech tagging the available ELRC - corpus developed at Ethiopian Languages Research Center has been used. Statistical POS tagging models have been developed for Amharic based on this corpus. for English, publically available opensource POS tagging models have been adopted [47].

## 4.2 Software Tools & Techniques

Using the Linux Operating System as the main research environment, state-of-art Statistical machine translation toolkit [48] has been used for this study. Ubuntu 18.04 LTS has been used, as an operating system since it is the most suitable for Moses SMT toolkit and other available NLP toolkits used in this study.

HornMorpho [49] latest release version 3.0, is a Python program that analyzes Amharic, Oromo, and Tigrinya words into their constituent morphemes (meaningful parts) and generates words, given root or stem of the word and a representation of the word's grammatical structure. It is the only open-source tool for Amharic that changes the surface form of a word to its morphemes and linguistic features.

A multi-threaded implementation extension for HornMorpho has been developed using Python. This ability helps us to execute the finite state rule-based HornMorpho as different programs on every eight cores of our CPU simultaneously. This has been achieved by running eight threads at a time by splitting the file to be analyzed into eight as equal to the number of cores our computer has. Finally, the output of each core is combined into a single file, which returns the

analyzed form of the words. This is done due to the slow nature of HornMorpho. As discussed here, below is a pseudo-code which shows how the multi-traded HornMorpho analyzer works.

```
Data: List of word to be Analyzed or Segmented by HornMorpho
Result: Analayzed/ Segemeted File
initialization;
Count no of threads;
while avilable do
    while word in a file do
        read word;
        run hornmorpho Segmenter/Analayzer;
        if Hornmorpho know the word then
            collect STDOUT segmented/analayed result to file;
        else
            go to the next word;
        end
    end
    merge each thread result to a single file;
end
```

**Algorithm 1:** Multi-threaded Hornmorpho word Segmentation or Analyzer

NLTK [50] which was created in 2001 as part of a computational linguistics course at the University of Pennsylvania serves as the basis of many research projects. It is now open-sourced and has been used to train POS tag models for Amharic.

spaCy v2.0 industrial-strength Natural Language Processing toolkit excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. Independent research has confirmed that spaCy is the fastest in the world [51]. The built-in available neural English POS tag models and tag-sets have been directly adapted for POS tagging in English.

Moses [48] has been used as a primary SMT toolkit for all the experiment on the translation side since it is open-source and supports integrating linguistic features to the translation model. There are a lot of SMT toolkits like Joshua [52], Phrasal [53], Asiya [54] and Jane [27] but Moses have built-in libraries for language modeling, word alignment, tuning and alternative confusion network decoding along with the phrase-based SMT.

mGIZA, a word alignment tool based on famous GIZA++ [55] was used since it has extended support for multi-threading in our octa-core experimental setup. It has allowed us to resume training and incremental training in power shortage times.

IRSTLM [56], a language modeling toolkit has been used since it has been successfully deployed with the Moses toolkit for statistical machine translation. The IRSTLM toolkit supports the distribution of n-gram collection and smoothing over a computer cluster, a language model compression through probability quantization, lazy-loading of huge language models from disk to permits efficient handling of language models with billions of n-grams on conventional low spec machines.

PYTHON, a high-level general-purpose programming language has been used along with the necessary libraries for pre-processing and data pre-processing, POS tag model development. Python is a programming language that lets you work more quickly and integrate your systems

more effectively [57]. Libraries such as NLTK and TreeTager helps us do more work in less time using python a major programming language.

Mendeley a free open-source reference manager and an academic social networking toolkit has been also used to manage references used in this study. With the Mendeley Reference Manager, you can easily organize and search your library, annotate documents and cite as you write and integrate to the LaTeXusing the bibliography BibLatex library.

## 4.3 Evaluation Technique

In any machine translation system, after a translation model is trained its performance will be evaluated using a human evaluation method or automatic evaluation method. Since the human evaluation method is tedious, error-prone, time-consuming and not efficient Bilingual Evaluation Understudy (BLEU) has been used to evaluate the performance of the system using a test data, which is an automatic evaluation method of MT models [58].

It is for evaluating the translation model built and measuring the quality of translations by using reference test data. Its an algorithm for evaluating the quality of test data which has been machine-translated from one natural language to another. BLEU was one of the metrics to achieve a high correlation with reference translation and remains one of the most popular automated and inexpensive metrics used in different researches for evaluation purpose. BLEU takes the n-grams from the candidate sentence and tries to match them in the reference test sentences. The more n-grams matched, the higher the candidate sentence's score is [58].

The score of a candidate translation is the maximum number of matched n-grams from the candidate divided by the total number of n-grams in the candidate. To find the BLEU score, first, the geometric average of the modified n-gram precisions, $p_n$, is computed using n-grams up to length $N$ and positive weights $W_n$ summing to one [58]. $c$ is the length of the candidate translation and $r$ is the effective reference corpus length. Then finding the brevity penalty $BP$ is computed as shown in Formula (4.1)

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \tag{4.1}$$

BLEU score is found as shown in Formula (4.2) after the brevity penalty is computed.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4.2}$$

To build a translation model by incorporating linguistic information, each token should be expressed with at least one feature which can best explain it from the natural language. In this study the token of the word itself is incorporated with POS tags, Lemma of the word and morpheme segmented representation of the word to improve the translation accuracy for Amharic - English, and English - Amharic translation.

Each candidate feature needs to be evaluated before it is used to create a translation model as being an extra feature to the token. To integrate a linguistic feature each token has to be gone in the process of morphology segmentation and analysis, in which the tools to do so have to be evaluated. For example, word lemma coverage has to be evaluated and POS tagger accuracy

has to be evaluated. Before we see how each of these components merge to form a collective improvement to Amharic - English translation, we need to clarify how our system is built.

## 4.4 Architecture of the System

The system architecture is designed to represent the process of factored machine translation, a way to incorporate linguistic information to a statistical translation. Figure 4.2 shows the architecture of the proposed phrase-based and factored statistical Machine translation. As can be seen from the diagram the system takes in the parallel and monolingual corpus as input. In the case of factored translation, the input parallel corpus contains a factored word with all the necessary features by using HornMorpho and the best tagger described in section 4.6. These sentences are used to develop language (monolingual corpus) and translation models(parallel corpus) respectively as discussed section 2.3.2.1.

The architecture in Figure 4.2 shows one side factored translation from Amharic to English, for the other vice verse translation there is no shortcut, its do it again by changing the source from Amharic to English and the target as English. It can also noted from the diagram POS tagger, morphological anlayzer and segmenter is needed on the training and tuning set, not on the referene test-sets. See further details of the overall building blocks of the experiment are discussed in Chapter 5 Section 5.1.

## 4.5 Morphological Analysis

One of the short-comings in traditional surface word approach in statistical machine translation is the poor handling of morphology. Each word form is treated as a token in itself as if it is independent of others. This means that the translation model treats, say, for example, the word "ቤቶች" completely independent of the word "ቤት". Any number of "ቤት" in the training corpus does not add any knowledge to the translation of "ቤቶች".

In sparsed under-resourced languages like Amharic, while the translation of "ቤት" may be known to a statistical model, the word "ቤቶች" mostly be unknown and the system will not be able to translate it.

Due to the limited morphological inflection in English, these extreme cases do not show up as strongly as in Amharic. But for Semitic languages like Amharic, it does constitute a significant problem due to morphological richness of the language.

Thus in this study, we preferably choose to model translation between Amharic and English on the level of lemmas, and thus pooling the evidence for different word forms like "ቤቶች", "ቤቷ", "ቤቱ", "ቤታቸው", "ቤቶቻቸው", "የቤቱ", "እንደቤታቸው" that derive from a common lemma "ቤት".

To achieve this, we translate Amharic lemma to English lemma and combine Amharic morphological information separately on the output side to ultimately generate the output surface words in addition to surface word translation. This is further discussed later in this chapter on Section 4.4. But here we discuss the processes we followed and toolkits we have used for Amharic and English morphological analysis plus segmentation for factored data preparation.

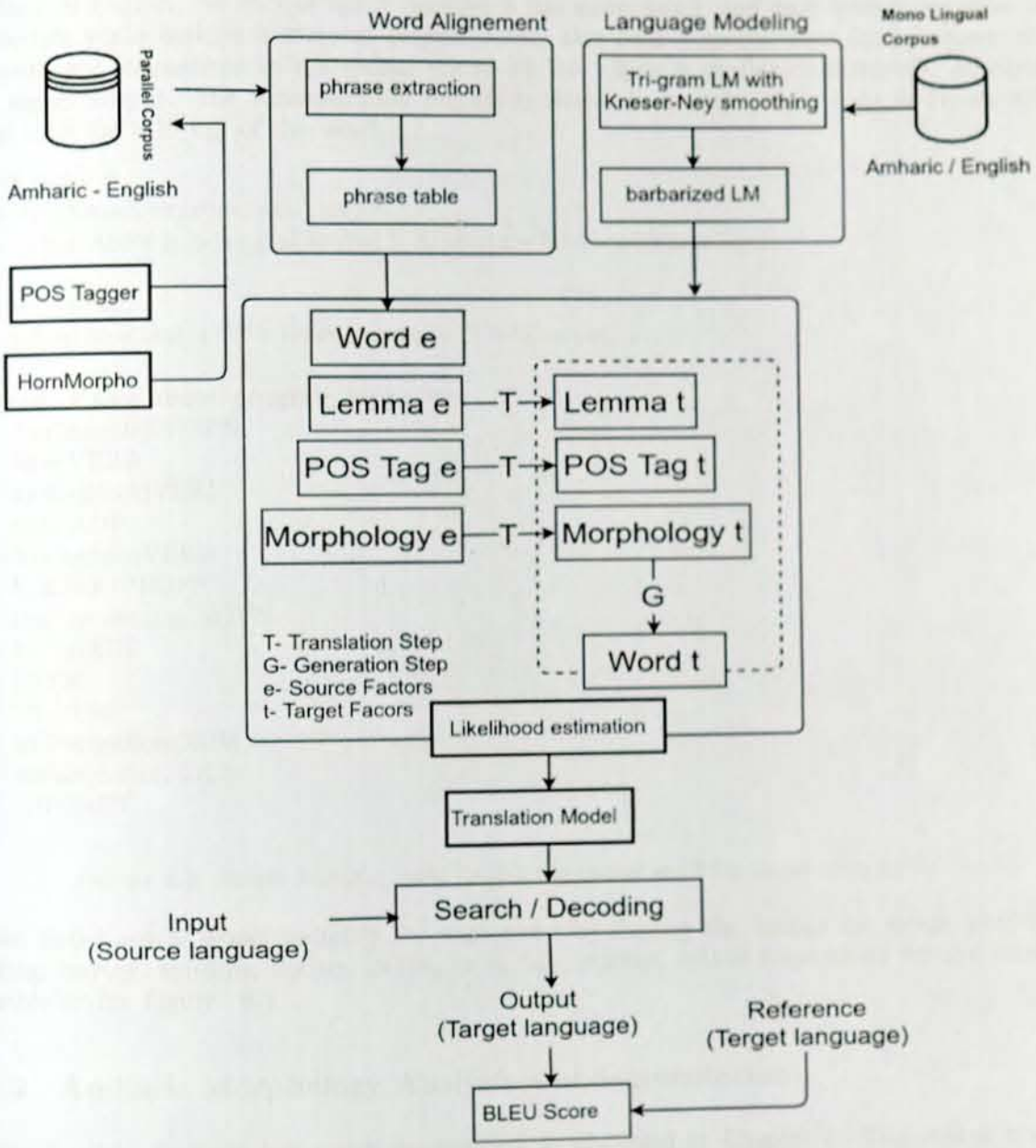


FIGURE 4.2: General architecture of a proposed Factored SMT

### 4.5.1 English Lemmatization

English morphological analyzer was done using spaCy NLP toolkit [51], in which the lemmatization data is taken from WordNet. WordNet® [59] is a large freely and publicly available lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. WordNet superficially resembles a thesaurus, in that it groups words based on their meanings.

Even though both NLTK [50] and spaCy [51] have lemmatizers over the WordNet® lexical

database of English, we choose spaCy because it has more speed, and easy later integration in our scripts while making a factored corpus. spaCy also adds a special case for pronouns: all pronouns are lemmatized to the special token -PRON- which is similar to ELRC/WIC Amharic POS tagger corpus. The lemmatization process is shown in a very simple code in Figure 4.3 along with the POS tag of the word.

```
import spacy
nlp = spacy.load('en_core_web_sm')
doc = nlp(u'Apple is looking at buying U.K. startups for $1 million dollars.')

for token in doc:
    print(token.text + "|"+ token.lemma_+ "|"+token.pos_)
```

The out of this above program looks like:

```
Apple|apple|PROPN
is|be|VERB
looking|look|VERB
at|at|ADP
buying|buy|VERB
U.K.|u.k.|PROPN
startups|startup|NOUN
for|for|ADP
||SYM
1|1|NUM
million|million|NUM
dollars|dollar|NOUN
.|.|PUNCT
```

FIGURE 4.3: Simple WordNet based English lemmatizer and POS tagger using spaCy

It can noted spaCy works perfectly for each word by finding the lemma for words likel is, looking, buying, startups, dollars] to [be, look, buy, startup, dollar] respectively for the above example in the Figure 4.3.

### 4.5.2 Amharic Morphology Analysis and Segmentation

Unlike English, Amharic has a rich morphology as discussed in Chapter 3. This makes morphological analysis very important for Amharic because it is practically impossible to store all possible words in a lexicon like WordNet® [59] for English, and many words have close to 0 probability of occurrence in any given corpus. This becomes obvious in the context of machine translation from Amharic to English, where the correspondence between words in Amharic will often be one-to-many.

The Amharic word **"ባይሰረዝላትም"**, for example, could be translated as "even if it is not erased for her". While a system for processing English could include all of the English words in the translation (even, if, it, isn't, erased, for, her) in its lexicon, an Amharic system that includes all words such as **"ባይሰረዝላትም"** is difficult [49].

HORNMORPHO [49] is a Python program that performs morphology analysis and generation of words in three languages of the Horn of Africa: Amharic (**አማርኛ**), Oromo (Afaan Oromoo,

Oromiffa), and Tigrinya (ትግርኛ). It is developed as part of the Processing Languages of the Global South (PLoGS) project at Indiana University, dedicated to developing computational tools for under-resourced languages.

HORNMORPHO analyzes words into their constituent morphemes which is the most important thing to integrate to our Factored Corpus. For Example, given a word we mentioned above "ባይሰሪዝላትም" to analyze it will return an output result as shown in Table 4.2:

| Key | Value |
|---|---|
| POS: | verb, root: <sr_z>, citation: ተሰረዘ |
| citation: | ተሰረዘ |
| subject: | 3 sing mas |
| object: | 3 sing fem prep: ል |
| grammar: | imperfective, passive, negative |
| conj prefix: | bI conj suffix: m |

TABLE 4.2: Sample Hornmorpho Output

which in this case the most information at this point in Table 4.2 is the citation/lemma form "ተሰረዘ" which is returned from the surface form "ባይሰሪዝላትም".

On the hand, HORNMORPHO also does segmentation, which prints out segmentation of the input word rather than its root-like the analyzer does. In the segmentation, the stem is surrounded by braces, and prefixes and suffixes are separated by hyphens. For Amharic, morphemes are described in terms of the grammatical features they represent; these descriptions appear in parentheses, with alternative interpretations for a morpheme separated by '|' [49]. For example, a surface token "የምንጨርሳቸው" is segmented as:

```
v:yem_(rel)-'n(sb=1p)-{Crs+1e2_3}(imprf)-ac_ew(ob=3p)
yem_-'n-{Crs+1e2_3}-ac_ew
```

The output can be further post-processed ignoring the unnecessary information and returning the text to Amharic from its a romanized Sera [36] representation as "የም_:እን:{ጭርስ+1አ2_3}:አች_አው" further dropping gemination, and inserting the "አ" vowel to the "CCC" Amharic root "ጭርስ" a final word "የም-እን-ጨርስ-አች_አው" has been used for the factored data as a morpheme segmentation in addition with the lemma from the Hornmorpho analyzer.

#### 4.5.2.1 Pre-processing of Hornmorpho Input

HORNMORPHO version 2.5 has segmented 46,373 of our word types out of 73,907 which is 62.74% in which this result is not bad for under-resourced languages like Amharic but to use the morpheme segmentation as a feature in factored corpus it not enough. This makes it hard to use morphologically segmented words as a feature for the factored data. So we consulted HornMorpho for a developmental version soon to be released @4.0 Alpha on their Github page.

We have done some minor *pre- and *-post-processing on the input and output of Hornmorpho respectively so as to speed up and increase token coverage since it has a limited number of rules due to its rule-based nature. Firstly we have changed our texts to a romanized form using Ge'ez to Sera representation mapping table [36]. In this process, we limit the Hornmorpho to a maximum number of analyses to consult to one, no grammatical analysis and disabling post-processing on the form plus enabling auto-guessing to use probabilistic guessing for words

where Honmorpho has no rule to consult. This has helped us to segment more than 82% of the corpus 60,930 to be exactly as shown in Figure 4.4.

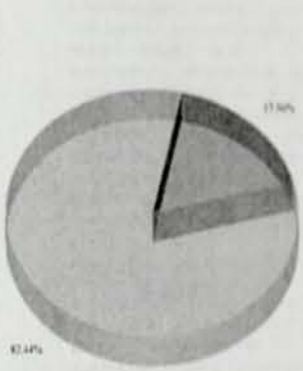

FIGURE 4.4: HornMorpho word segmentation Coverage

#### 4.5.2.2 Post-processing of Hornmorpho Output

To convert HornMorpho output such as "v:yem_(rel)-'n(sb=1p)-{Crs+1e2_3}(imprf)-ac_ew(ob=3p)" to their correct form "የም-እን-ጨርስ-አች_አው"we have :

- changed the romanized text back to sera.
- removing grammar information for each constituent.
- remove geminations and other phonetical markers since they do not matter in translation

A sample post processed lemma and morphological information output have been shown in Figure 4.5. For words that has no citation or segmentation format we use the surface form itself as a lemma and morphological feature.

## 4.6 POS Tagging

In corpus linguistics, Part of speech tagging, also called grammatical tagging, is the process of marking up a word in a corpus as corresponding to a particular part of speech, based on both it's definition, as well as it's context through relationships with adjacent and related words in a phrase, sentence, or paragraph.

POS tagging is required in this research to include POS information to the factored corpus for both English and Amharic. Tagging the English corpus was easy compared to Amharic since there were internationally available open source tools to do so. But for Amharic, though there there were internationally available open source tools to do so. But for Amharic, there was no model or toolset to are a number of papers reported on Amharic POS tagging, there was no model or toolset to adopt.

```
ከአንቀ: አንቀ|ከ-አንቀ-ኡ
ምሥክሮቻቸውን: ምሰክር|ምሰክር-አች-አቸው-ን
ከኅላማው: አለማ|ከ-አለማ-ኡ
ይንድለኛል: ገው|ይ-ገው-ድ-ኧ-ው-አል
በማዕዘን: ማእዘን|በ-ማእዘን
ሕዝቢና: ህዝብ|ህዝብ-ዋ-ና
እመሠክራለሁ: መሰክር|እ-መሰክር-አለሁ
ኃጢአቱን: ሀጢአት|ሀጢአት-ኡ-ን
ማዕድ: ማእድ|ማእድ
አይታገሥም: ችታገስ|አይ-ችታገስ-ም
ዕቃውን: እቃ|እቃ-ኡ-ን
ያንርፋ: አገሙር|ይ-አገሙር-ኡ
በጎሎችሽን: በአል|በአል-አች-ሽ-ን
ንጹናውንም: ንጹና|ንጹና-ኡ-ን-ም
ሕይወትም: ህይወት|ህይወት-ም
ኃጢአቴን: ሀጢአት|ሀጢአት-ኡ-ን
ሌሞንን: ሄሞን|ሄሞን-ን
በግጥን: ሳጥን|በ-ሳጥን
ሥራምችህ: ስራ|ስራ-አች-ህ
ውኃን: ውሀ|ውሀ-ን
መግሪያምቹ: መሳሪያ|መሳሪያ-አች-ኡ
ሠርተናልና: ሰርት|ሰርት-ኧን-አል-ና
ንግንም: ንግ|ንግ-ን-ም
አልንዲኧም: ገመአ|አል-ገመአ-ኡ-ኧ-ም
ለኃጢአታችን: ሀጢአት|ለ-ሀጢአት-አችን
የሕጉ: ህግ|የ-ህግ-ኡ
የተመሠክረለት: ተመሰክር|የ-ተመሰክር-ኧ-ል-ኧት
```

FIGURE 4.5: Sample Lemma and Morphology of Amharic Words.

### 4.6.1 POS for English

The English part-of-speech tagger uses the OntoNotes 5 [60] version of the Penn Treebank [61] tag set by adopting the simpler Google Universal POS tag sets using Spacy[51] 2.0 neural models. It is fast,easy to use and is the best tagger with 97.20% accuracy, the best in the open-source world now.

### 4.6.2 POS for Amharic

In this research, we have compared and contrasted previous Amharic POS tagging ways reported, by following the same ways as the corresponding authors and checking the accuracy on a randomly selected test set.

The corpus used in this study for Amharic is known as ELRC - corpus developed at Ethiopian Languages Research Center has 31 tag-sets where each word manually tagged by linguists. This same corpus has been referred to as WIC - Corpus referring from where the content is taken from, Walta Information Center which is a news agency [47] [8]. It consists of 210,000 annotated tokens of Amharic news documents. The corpus contains a total of 8,051 relatively long sentences due to the news nature of the document. We have randomly divided these sentences into a training and test set, 95% - 5% ratio respectively, 7649 sentences for train and 402 for testing.

The ELRC tagset used to tag the corpus is based on 11 basic tags, most of which have further been refined to provide more linguistic information, thus increasing the tagset to 31. For example, the tags for Nouns are VN (Verbal Noun), NP (Noun with Proposition), NC (Noun with Conjunction), NPC (Noun with Preposition and Conjunction) and N (for any other Noun). There are similar patterns for Verbs, ADJectives, PRONouns and NUMerals. Additional tags under the verb category are tags AUX (for AUXiliary) and VREL (for RELative Verbs). Numerals are divided into cardinals and ordinals represented by the NUMCR and NUMOR tags. The

rest of the tags are PREP for prepositions, CONJ for conjunctions, ADV for adverbs, PUNC for punctuation, INT for interjection and UNC for unclear (difficult to classify) [47]. As shown on a sample of ELRC/WIC corpus in the Figure 4.6 the Word and the tag are separated by "|".

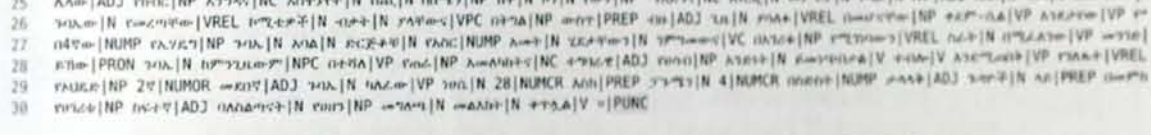

FIGURE 4.6: Sample ELRC/WIC manually tagged Amharic POS Corpus

#### 4.6.2.1 POS Experiment Descriptions

The fact that Amharic has no POS tagging tool or available model, we have developed different types of POS taggers using the same traning and tes set explained in Section 4.6.2. We have compared and contarsted the performance of each tagger with reported previous results if available and also against themselves.

The simplest tagger that can serve as a baseline in Amharic part of speech tagging is to tag all new tokens as N - NOUN, which is the most frequent tag in the ELRC corpus. This achieves an accuracy of about **33.23%**. This is too low to be used as a baseline as most algorithms have much higher accuracies reported. All the algorithms applied in this paper achieve much higher accuracies than **80%**.

All Experiments conducted and listed in Table 4.3 are developed using the same training and test sets. Comparisons and contrasts have made on studies conducted on the same ELRC/WIC corpus, are listed below. In all the taggers listed Tagger have been trained first and evaluation has been performed before being used to tag input.

We have used NLTK to train most of the POS Tagging trainings using the following sequence of pseudo-code for TNT, N-gram taggers(4-gram and 5-gram), Affix tagger, Brill tagger and Classification based taggers with Nive-Bays algorithm.CRF++, SVMTool, MALLET was used to train POS tagger based on $CRF_s$, Support vector machine and and Maximum Entropy respectively for experiments other than those conducted using NLTK.

LISTING 4.1: Pseudo-code of Taggers trained

| Exp. | Algorithm | Best Result yet | Our Results | Remarks |
|---|---|---|---|---|
| 1 | TnT | 83.49 | 80.46 | Reported by [62] |
| 2 | 4-gram | - | 79.23 | ref (A) |
| 3 | 5-gram | - | 79.01 | ref (B) |
| 4 | Affix Tagger | - | 51.02 | with prefix length 1 |
| 5 | « | - | 47.93 | with prefix length 2 |
| 6 | « | - | 45.01 | with prefix length 3 |
| 7 | « | - | 32.01 | with suffix length 1 |
| 8 | « | - | 55.17 | with suffix length 2 |
| 9 | « | - | 59.43 | with suffix length 3 |
| 10 | Affix Tagger Backoff* | - | **60.62** | ref (C) |
| 11 | Nive Bayes | | 82.94 | |
| 11 | TNT Backoff* | | 85.56 | ref (D) |
| 12 | Maximum Entropy- MaxEnt | 87.87 | 84.58 | Reported by [8] |
| 13 | Support Vector Machines - SVM | 93.50 | 82.26 | |
| 14 | Conditional Random Fields -CRFs | 74.00 | 80.01 | |
| 14 | Brill Tagger | 87.41 | 85.20 | Reported by [37] |
| 15 | Nive Bayes Backoffed* | - | **90.02** | ref (E) |

TABLE 4.3: Amharic Part of Speech tagging Experiments - Average Accuracies (in %)

Note: The results herein are **overall accuracies** combination of both known and unknown words.

(A) : Experiment done to check the effect of higher level N-gram(4) orders.

(B) : Experiment done to check the effect of higher level N-gram(5) orders.

(C) : Combination of Affix based taggers with 1 prefix and 3 suffix of a word.

(D) : Combination of TnT with best Affix tagger.

(E) : Combination of Nive-Bayes with best Affix tagger on Number 10,Regexp Tagger and Default Tagger.

```
load corpus reader and tokenizers;
import defult taggers;
load modeling;

Read trainging and test data;
WHILE no error on the data:
        Tokenize sentences and words
                choose algorithm
                train tagger on train data;
END WHILE;
Evaluate tagger model by test data:
if a better model:
        Save model;
```

First experiment TnT, the short form of Trigrams'nTags, is a very efficient statistical part-of-speech tagger that is trainable on different languages and virtually any tagset. TnT uses a second-order Markov model to produce tags for a sequence of input.

Higher-level N-gram tagger chooses a token's tag based on its word string and the preceding n word's tags. Both 4-gram and 5-gram have tended to be the poorest of all taggers used in

this study, accuracy less than 80%. This is due to the limited nature of manually tagged text in the WIC corpus.

Affix Tagger, a tagger that chooses a token's tag based on a leading or trailing substring of its word string. It is important to note that these substrings are not necessarily "true" morphological affixes that could lead to change in POS tag. In particular, a fixed-length substring of the word is looked up in a table, and the corresponding tag is returned. Affix taggers are typically constructed by training them on a tagged corpus.

Though there has not been a study report on Affix taggers for taggers previously, combining a single prefix substring and there suffix constituents tends to work more than 60% for Amharic. Affix taggers are also easy to back off with other statistical taggers to improve tagging efficiencies.

Classifier-based Tagger uses a classifier algorithm (specifically Naive Bayes in our case) to choose the tag for each token in a sentence. The feature set input for the classifier is generated by a feature detector function:

```
feature_detector(tokens, index, history) -> featureset
```

Where:

- tokens is the list of unlabeled tokens in the sentence;
- index is the index of the token for which feature detection should be performed;
- history is list of the tags for all tokens before index.

This classifier based taggers also works at a fine margin especially when backed off with Affix taggers, of more than 85% accuracy.

The best tagger adopted for this study is NiveBayes based Classifier tagger backed off with multiple other taggers, namely Affix tagger, regular expression tagger, and Default "Noun" tagger is the only tagger with an accuracy of more than 90%. We set the "cutoff_prob" to 66.5% so that this Naive-bays classifier will fall back on its backoff tagger if the probability of the most likely tag is less than "cutoff_prob". Even though this takes a lot of days to train, the final model is used as a python pickled file throughout this study. The complete code used to develop a tagger with NLTK is listed in Appendix B.

As shown Appendix B code snip, Regexp Tagger is used to assigns tags to tokens by comparing their word strings to a series of regular expressions. This matching word has always had single POS tags due to the linguistic nature of the language as e.g. words in Amharic which end "ዎች" or "ኦች" are categorized as Nouns(N). Other words like "እንደሆነ", "መሆኑ" and "ሲሆን" are Auxiliaries(AUX).As shown in the above code snip, Regexp Tagger is used to assigns tags to tokens by comparing their word strings to a series of regular expressions. This matching word has always had single POS tags due to the linguistic nature of the language as e.g. words in Amharic which end's "ዎች" or "ኦች" are categorized as Nouns(N). Other words like "እንደሆነ", "መሆኑ" and "ሲሆን" are Auxiliaries(AUX).

## 4.7 Factored Data Preparation

In factored translation, a direct surface to surface word data is not usable unless it is tagged. We create a factored representation of the data to build factored translation systems as explained by Koehn and Hoang [4]. In this representation, the data is not only represented by surface

forms but also a combination of lemmas, POS and/or Morpheme segments. These three features are used as factors for the factored translation systems as they are reflected in the training data. In the factored data, factors are separated by a "|" symbol and we represent the data as Surface_Form|Lemma|POS and or Morpheme_Segmentation continuing to as many combinations they have and as many features, the language could have. Table 4.4 illustrates examples of all these representations.

| Representation | English Data | Amharic Data |
|---|---|---|
| Surface | horses | በፈረሶች |
| Lemma | horse | ፈረስ |
| POS | NNS(noun, plural) | NP(Noun Phrase) |
| Tagged Form | horses\|NNS | በፈረሶች\|NP |
| Factored Form | horses\|horse\|NNS | በፈረሶች\|ፈረስ\|NP\|በ-ፈረስ-ኦች |

TABLE 4.4: Amharic vs English factored data representations

As shown in table, we don't use morpheme tag information for English like we do for Amharic because English has less inflectional morphologies, and even if it does it is easy to represent morphemes in the POS tags. Penn-Treebank [61] tag set has a maximum of 56 tag-sets to handle such problems. So unlike Amharic POS corpus in English:

- "horses" has a tag of "NNS"
- "horse" has a tag of "NN"

due to the morpheme "s" which is a plural marker. This is adapted from simpler Google Universal POS tag set for this study. A sample of factored corpus data is shown in Figure 4.7.

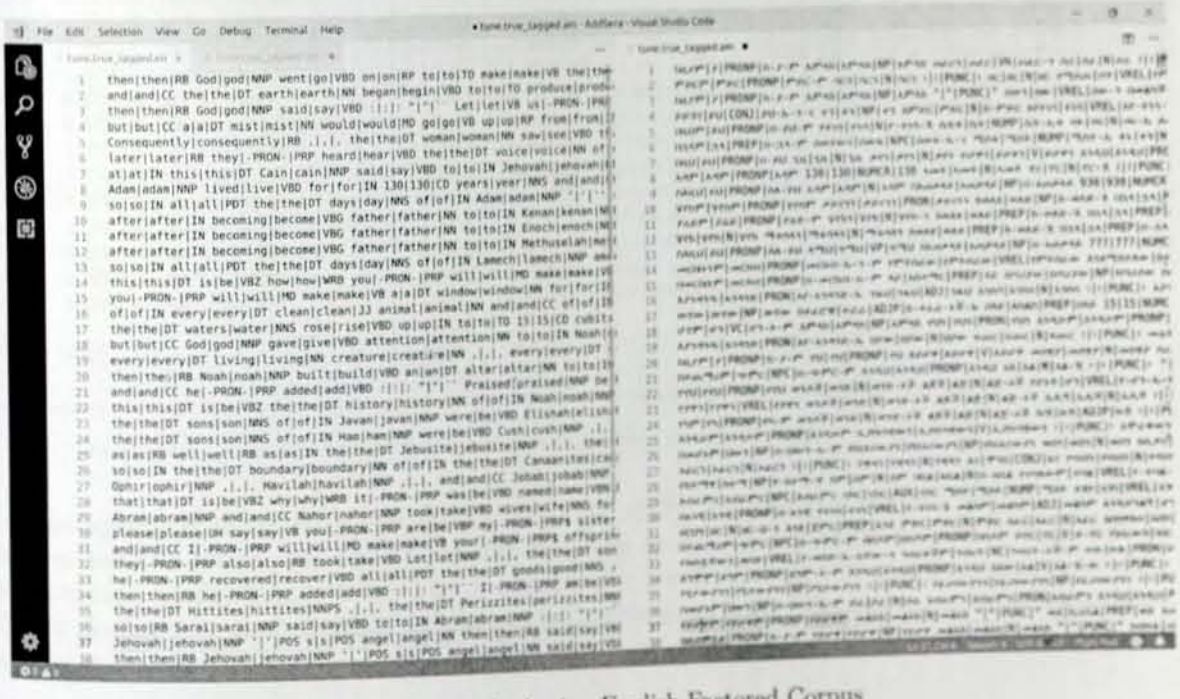

FIGURE 4.7: Amharic - English Factored Corpus

### 4.7.1 Creating Language Model

As explained in Section 2.3.2.1.1 language model is only needed for the target language, but since we do a bidirectional translation we create language models for both Amharic and English using IRSTLM toolkit [56]. We use 3-gram language models that perform better than the others as explained later in Chapter 5 Section 5.3.5 later in this. We create the language models for our different systems with Kneser-Ney smoothing algorithm [63] and binarize the models to shrink the size for memory load. The binarization step is done using the scripts provided with Moses toolkit [48].

For our four different systems, three language models are created. These are created respectively from the POS tags, lemmas and surface forms from the data itself. We haven't created a language model for morpheme segments because we have not used the segmentation in the translation step, it has only be adopted for the generation of POS and Surface word. The number of translation steps should be similar to the number of language models in SMT else the language model will not have any effect for fluency. Some examples from these three language models are shown in Table 4.5.

| Surface Word Lang. Model | | Lemma Lang. Model | | POS tag-set Lang. Model | |
|---|---|---|---|---|---|
| 771564 | -0.5588497 አውሬውም አመገዝራቸን ይጠሏታል | 575033 | -0.7526794 በድኛ ቃጠል ሀይት | 8845 | -3.4554331 N PRON PRONC |
| 771565 | -0.8784163 ያባዘባቢታል ፤ ራቁቷንም | 575034 | -1.0424411 ደህ አሳት ሀይት | 8846 | -3.3058133 PREP PRON PRONC |
| 771566 | -0.5588497 ፤ ራቁቷንም ያስቀሯታል | 575035 | -0.8693204 ኪሩበል አሳት ሀይት | 8847 | -2.7426288 VP PRON PRONC |
| 771567 | -1.6546584 ያሳነሰል ! ፍርሰሽ | 575036 | -1.551434 ነደፀ አሳት ሀይት | 8848 | -1.4898374 NUMCR PRON PRONC |
| 771568 | -0.87839574 ሙታንን ፤ ታላላቆችንና | 575037 | -0.71352774 ኪሩበል ጸኝ ሀይት | 8849 | -3.251834 N CONJ PRONC |
| 771569 | -0.5588497 ፤ ታላላቆችንና ታናናሾችን | 575038 | -0.7281417 ከአንገላህ ህዝቦችን አንዳጦስት | 8850 | -2.488884 PUNC CONJ PRONC |
| 771570 | -0.7740108 ፤ የሙጽዐሩ ጥቅልሎችም | 575039 | -1.9018552 ዚን ፤ ሰማሀኖም | 8851 | -2.865617 PRONP CONJ PRONC |
| 771571 | -0.85837877 ። ሙታን በጥቅልሎቹ | 575040 | -0.56074417 ፤ ኑግንና ማኑግን | 8852 | -2.2238114 V CONJ PRONC |
| 771572 | -0.87807 በጥቅልሎቹ ውስጥ በተጻፉት | 575041 | -1.0393707 ፀሉ አግ ፋንሶቸው | 8853 | -2.1814089 ADJ VPC PRONC |
| 771573 | -0.8779323 የቀድሞው ምድር አልፈዋልና | 575042 | -4.0370693 አሰቀር = የብርሃኔም | 8854 | -2.967117 N VPC PRONC |
| 771574 | -1.3554299 ከተማቸን ፤ በር፰ንና | 575043 | -0.81222683 ጀምር ከሰሰል ጥርት | 8855 | -1.8621614 VN VPC PRONC |
| 771575 | -1.6671972 የዘበሩ ደንጉዮች ያጌጡ | 575044 | -3.3024788 ሃዘር ፤ የ12ቱም | 8856 | -1.9578409 PRON VPC PRONC |
| 771576 | -1.756021 ፤ ሶስተኛው ከልቁሶን | 575045 | -0.71694124 አፕር ተማስ ከኢያስጸደ | 8857 | -0.7314622 PRONC VPC PRONC |
| 771577 | -0.8720206 በሮች 12 አንቀዖች | 575046 | -1.7007544 <s> አምስተኛ በርደናነ | 8858 | -2.7804815 PUNC VC PRONC |
| 771578 | -0.8756046 ጐኖኖም ብርሃን እንደሚያስተላልፉ | 575047 | -1.5107267 ፤ ስደስተኛ በርደኖን | 8859 | -2.459673 PUNC PRONC PRONC |
| 771579 | -1.6820328 <s> ከተማዋ የጸወይም | 575048 | -1.2882884 ፤ ስምንተኛ ቢረል | 8860 | -3.5431616 NP NC ADJPC |
| 771580 | -0.8756046 የጨረቃ ብርሃን አለስፈለጋትም | 575049 | -1.0791067 ፤ አስረኛው ክርስቶፈሰስ | 8861 | -4.6478027 N N ADJPC |
| 771581 | -1.1738088 ከባር ብርሃን ሰዋቷታልና | 575050 | -1.3297709 አስራ አንደኛ ያክስትና | 8862 | -3.8981937 VREL N ADJPC |
| 771582 | -1.1792204 ከብራዛውን ወደ እሷ ያመጣሉ | 575051 | -1.3492752 ደግም አሰላለም ይገግሰሉ | 8863 | -4.2809645 N VREL ADJPC |
| 771583 | -0.5588497 <s> መንፈሰና ሙሽራይቱም | 575052 | -2.6885705 ርክስ ሆን በርክሰት | 8864 | -4.655113 V PUNC ADJPC |
| 771584 | -2.1565943 ይላሉ ፤ የሚሰማም | 575053 | -0.7525688 ለመመስከር መልክ ልሂለስ | 8865 | -3.3149118 PREP VP NUMPC |
| 771585 | -1.1794136 ደበል ፤ የተጠማም | 575054 | -2.517058 እነዚህ ነገር የሚመሰክረው | 8866 | -3.5000157 PRON VP NUMPC |
| 771586 | | 575055 | | 8867 | |
| 771587 | \end\ | 575056 | \end\ | 8868 | \end\ |
| 771588 | | 575057 | | 8869 | |

TABLE 4.5: 3-Gram Amharic Surface Word, Lemma and POS tag-set Language models

We have specified three language models for this study, besides the regular language model based on surface forms, we have a second language model that is trained on POS tags and a third language model that is trained on a lemma of a word. The part-of-speech language model includes preferences such as that determiner-adjective are likely followed by a noun, and less likely by a determiner:

```
-0.192859       dt jj nn
-2.952967       dt jj dt
```

These combinations of language models are used just like normal surface form model, during the decoding process, not only tokens(በፈረሶች) but also part-of-speech tag(NP) and lemma(ፈረስ) are generated. Translation models will have better improvements by preferring a sequence of POS tags where words have not been seen next to each other before, so surface form the language model has very little to do in such scenarios. As shown in the foregoing N-gram count of Table 4.5 the part-of-speech language model is aware of the count of the pronoun (PRON) involved and prefers a verb phrase(VP) before a numeric conjugate (NUMPC) over others in Amharic.

## 4.8 General Experiment Workflow

As it can be noted from the Experimental setup adding linguistic feature to a corpus and then performing translation by performing word alignment, training translation model, tuning, and the evaluation is tedious and more likely will lead to error.

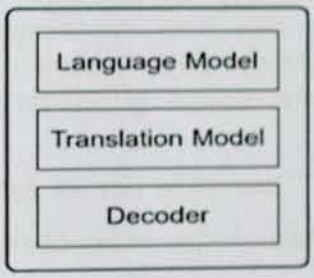


FIGURE 4.8: General workflow and outline a Machine Translation system

Apart from the separate process of POS tagging and Morphology segmentation and analyzers, we have written a shell command to combine all the tasks shown in Figure 4.8 all in one task. This has helped us to execute the script by changing only simple parameters at the time. A sample shell program used has been shown in the Appendix.

# Chapter 5

# EXPERIMENT AND DISCUSSION

This chapter presents the results of the different sets of experiments carried out for this study. The necessary detail of corpora that are used along with factoring mechanisms during the experiments is presented in the previous Chapter. This chapter starts with the experimental setup, followed by presenting and discussing the improvements in translation quality based on different experiments. These experiments have been conducted by combing Lemma and POS tag to the surface word one at a time. After the each linguistic feature was studied, a combination of each linguistic was studied by adding one linguistic feature at a time to its preceding experimental setup.

The five major experiments conducted for this study are:

1. Surface(word-based) experiments (baseline experiment)
2. Surface and POS tagged experiments
3. Surface and Lemma experiments
4. Surface, Lemma and POS tagged experiments
5. Surface, Lemma , POS tagged and Morpheme's experiments

These five experiments were conducted by starting from the baseline translation, taking the surface word and creating a phrase-based statistical machine translation model bidirectionally. This experiment was used as a benchmark for the next experiments where a factored model was developed by using the surface word and each language feature, starting from Surface and POS, then Surface and Lemma, lastly Surface combined with Morpheme segmented experiments. Since English has little morphology, we have not conducted Surface and Morpheme combination on its own, rather we use the morpheme representation of Amharic as a generation model combined with all the other language features in the last experiment.

The Experiments was further extended not by combining each language feature to the surface form at a time but also by adding more language features to the baseline incrementally. This makes a combination of Surface, POS tag and Lemma as another experiment and lastly the addition of morpheme segmentation combined with all the available language features as the last experiment. All these experiments were conducted bidirectionally to study the effect of each language feature to translate to Amharic or translate from Amharic.

The chapter concludes by comparing the results of each translation model and analyzing the translation quality of the generated translated output using the BLEU evaluation adopted for this study.

## 5.1 Training Models

Training is the most important step where the systems learn how to translate by estimating the parameters of the statistical translation model with the use of parallel corpus as discussed in Chapter 2 Section 2.3.2.1.1.

Translation models are trained by following the SMT pattern of computing word-alignments first, extracting and scoring phrase tables and then creating reordering tables. The training set is selected randomly from the parallel corpus before factored data preparation, so translation models are unique and recreated for each data-set based on the incorporated linguistic feature in the training set. Moses toolkit [18] is used to train the systems and create translation models.

To build the system we have prepared a factored data for the training-set and tuning-set as explained in Section 4.7. The correctness of this factored data is dependent on the HornMorpho coverage, a morphology analyzer for Amharic as well as the accuracy of our POS tagger, where each is discussed in Section 4.6.2.1 and 4.5.2. A sample of our first sentence in the training data is shown in Figure 5.1.

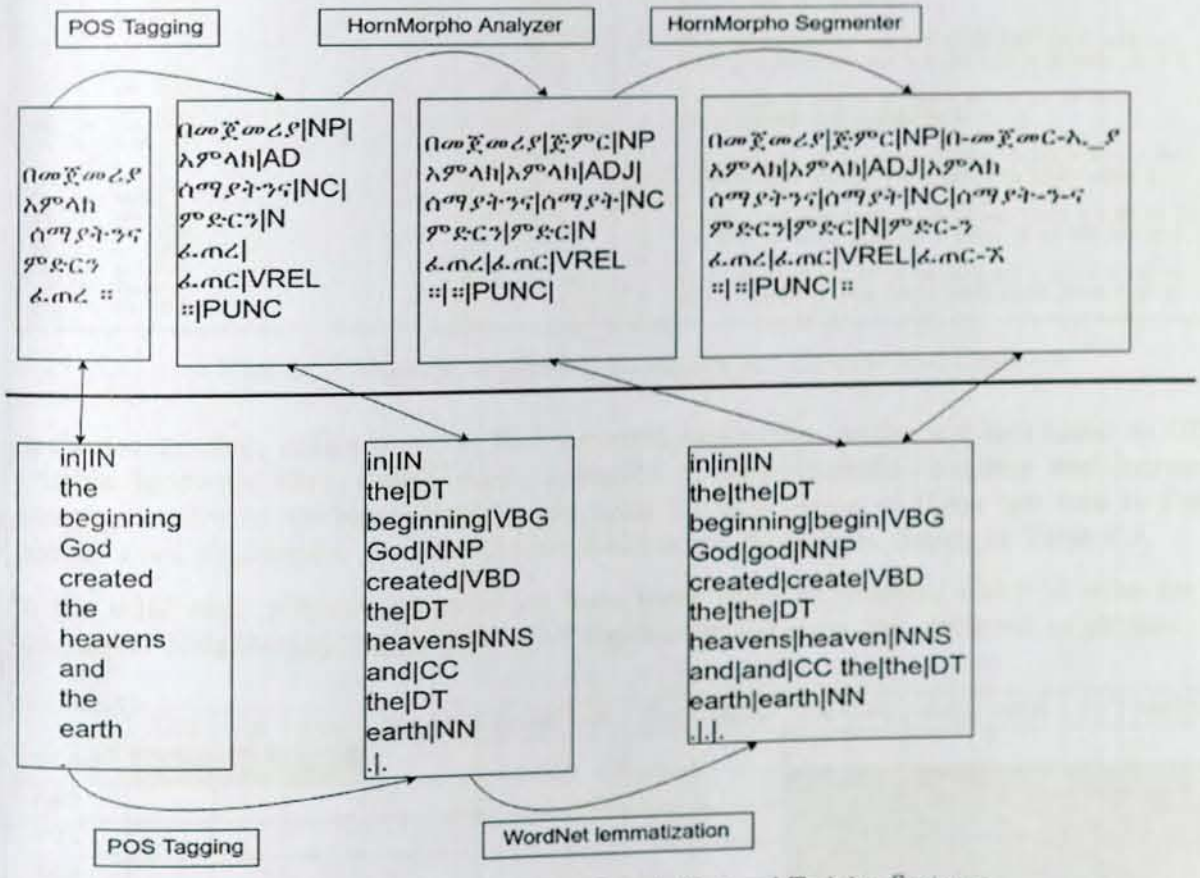


FIGURE 5.1: English - Amharic Sample Factored Training Sentence

### 5.1.1 Training Translation Systems

Training is the most important step where the systems learn how to translate by estimating the parameters of the statistical translation model with the use of parallel corpus as discussed in

Chapter 2 Section 2.3.2.1.1. Here, translation models are trained by computing word-alignments, extracting and scoring phrase tables and creating reordering tables. The training set is selected randomly from the parallel corpus, so translation models are unique and recreated for each data-set. Moses toolkit [18], is used to train the systems and create the translation models.

In training primary task is to create vocabulary files for both Amharic and English, these contain word identifiers (integers), words and word counts in the parallel corpus. With the help of these vocabulary files, each word in the parallel corpus is represented with its integer identifier) and a sentence-aligned corpus file is created with the word identifiers. Table 5.1 includes both examples of vocabulary files and sentence-aligned corpus file. In the vocabulary files, the first column is the word identifier, the second column is the word itself and the third column is the count of that word in the corpus. In sentence-aligned corpus file, a sentence pair is shown in three lines where first is the frequency of the sentence, second and third are the sentences from both languages where words are represented with their identifiers from the vocabulary files.

| Sample segments from vocabulary files | | | | | | | | Sample editorialized sentence-aligned corpus |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | UNK | 0 | 1 | 1 | UNK | 0 | 1 |
| 2 | 2 | ። | 28634 | 2 | 2 | the | 44371 | 9 2 922 42 1525 2 218 5 2 128 4 |
| 3 | 3 | ፣ | 17253 | 3 | 3 | . | 39485 | 948 24 4350 771 12339 2 |
| 4 | 4 | ፡ | 11024 | 4 | 4 | , | 27653 | 1 |
| 5 | 5 | " | 6284 | 5 | 5 | and | 24693 | 93 2 128 34 11578 5 816 3 5 67 34 566 222 2 1607 6 2 |
| 6 | 6 | ለዶ | 5229 | 6 | 6 | of | 23844 | 1373 1370 49349 902 613 3 17084 133 1023 52062 13 3 5 |
| 7 | 7 | " | 5061 | 7 | 7 | to | 18773 | 1 |
| 8 | 8 | ውደ | 4025 | 8 | 8 | you | 11301 | 5 42 45 24 17 338 67 33 393 4 25 507 67 34 393 4 |
| 9 | 9 | ነው | 3990 | 9 | 9 | in | 9639 | 856 5 290 101 7 62 2 7642 68 2 |
| 10 | 10 | ሁሉ | 3627 | 10 | 10 | will | 9377 | 1 |
| 11 | 11 | ይህም | 3180 | 11 | 11 | a | 7686 | 108 16 42 182 16 2 393 34 152 3 5 42 171 7 3242 2 393 |
| 12 | 12 | ? | 2707 | 12 | 12 | he | 7581 | 78 48 24 5169 128 166 775 3 856 7643 13830 16597 2 |
| 13 | 13 | ነገር | 2659 | 13 | 13 | I | 7197 | 1 |
| 14 | 14 | እንዲህ | 2585 | 14 | 14 | for | 6815 | 42 205 2 393 4326 3 44 2 566 12 205 13172 4 5 67 34 5 |
| 15 | 15 | በውስ | 2505 | 15 | 15 | is | 6461 | 856 7643 21 38 26 123 1626 3 17041 20 21 403 26 123 1 |
| 16 | 16 | ነበረ | 2426 | 16 | 16 | that | 6436 | 1 |
| 17 | 17 | - | 2367 | 17 | 17 | “ | 6276 | 53 42 45 24 17 338 67 33 98 3654 382 2 377 3 5 112 67 |
| 18 | 18 | ጋር | 2063 | 18 | 18 | his | 6086 | 22 24 5 15497 34 6016 101 4 19923 20296 29628 7 62 2 |
| 19 | 19 | ነው። | 2054 | 19 | 19 | Jehovah | 5710 | |
| 20 | 20 | ግን | 2010 | 20 | 20 | they | 5381 | |

TABLE 5.1: Example segment of Vocabulary and Sentence level alignment

In the second step, mGiza is run to find the word alignments. mGiza is a tool based on GIZA++ avilable in moses SMT toolkit stack, extended to support multi-threading and incremental training. mGiza is run bidirectionally and takes the intersection of these two runs to find the correct word alignments. An example word-alignment diagram is shown in Table 5.2.

In the third step, phrases are extracted from these word alignments, which is done for both languages. Neighboring words, that occur together in the data, are extracted as phrases. Here

| | And | the | rain | poured | down | on | the | earth | for | 40 | days | and | 40 | nights | . |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ዝናቡም | ■ | ■ | ■ | | | | | | | | | | | | |
| ለ40 | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | |
| ቀንና | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | |
| ለ40 | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | |
| ሌሊት | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | |
| በምድር | | | | | | | ■ | ■ | | | | | | | |
| ላይ | | | | | | ■ | | | | | | | | | |
| ወረደ | | | | ■ | ■ | | | | | | | | | | |
| ። | | | | | | | | | | | | | | | ■ |

TABLE 5.2: Example word-alignment between sample English-Amharic sentence pair


FIGURE 5.2: A portion from the phrase translation table

we use the default maximum phrase length for Moses, which is seven. Each phrase pair is assigned alignment points, showing the number of word alignments in this pair. Then these pairs are sorted for both languages and scored with the calculated probability of that translation. In the end, a translation table is created containing all the extracted phrase pairs and their scores. A small portion from the translation table is shown in Figure 5.2. The four different scores separated by "|||" at the end of every line are; inverse phrase translation probability, inverse lexical weighting, direct phrase translation probability, and direct lexical weight [18].

At the end of the training step, a configuration file is created for the Moses decoder a "moses.ini" file. A file which specifies, the factors used, the translation tables, the reordering models, the generation models, language models used and the weights of all these models.

#### 5.1.1.1 Factored Training

Even though most the training steps discussed above are similar in factored translation systems, multiple phrase translation tables are created according to the translation factors given to Moses toolkit. we have represented features in our factored corpus as:

1. feature 0 - word
2. feature 1 - Lemma
3. feature 2 - POS tag
4. feature 3 - Morpheme segments

Moses accepts a set of factor lists as translation factors and creates a translation table for each element in the set. For example, if the translation factors are given as 0-0+1-1, then translation tables will be created from source factor 0 to target factor 0 and from source factor 1 to target factor 1. This means, from the factored training data, a translation table will be created for the first factors and another will be created for the second factor.

In this study, translation factors are given as 0-0+1-1+2-2, which means separate translation tables are created for surface forms (factor 0), lemmas (factor 1) and POS/Morpheme tags (factor 2). An illustrative example of how translation factors work is shown in Figure 5.3

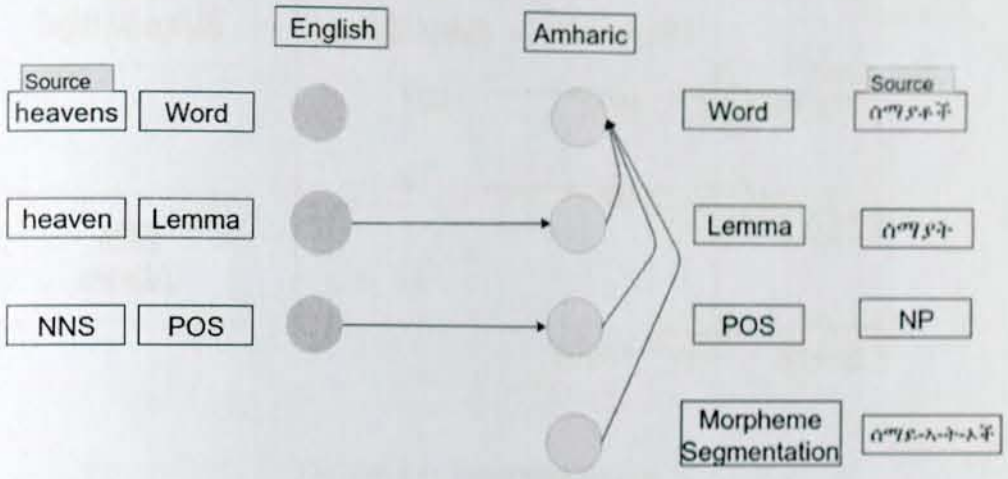


FIGURE 5.3: Sample Factored translation from English to Amharic

Factored systems are also used to train word alignments and create generation models for factors. In this study, in addition to word-level alignment, lemmas are also used to train the alignments as they are the smallest counterparts of the words and they occur more general than surface forms especially for Amharic due to its inflection in morphology from the same citation.

Generation models are used to decide which target side factors will be used to generate other target side factors. In this study, 0-1,3+1,3-0+3-2 is given as generation factors, which means three generation models are created, one for generating lemma and morpheme tags from a surface form, one for generating a surface form from a pair of lemma and morpheme tags and one generating POS from morpheme tag.

#### 5.1.1.1.1 Why do we generate POS from morpheme tag?

Because we have seen in the POS tagging experiments in Chapter 4 Section 4.6.2.1 Affix taggers work fine by considering one prefix and three suffix constituents, thus morpheme tag constituents(factor 3) here can help generate POS tags for Amharic. This has proven to generate little translation accuracy in experiments listed late in this chapter. Figure 5.4 shows an illustrative example of how generation factors work.

Figure 5.4 shows an illustrative example of how generation factors work.

### 5.1.1.2 Tuning

Moses uses the created models with different weights. These weights are recorded in the configuration file which is created after the training step. However, in the configuration file, these weights have default values and they need to be optimized. This optimization is done at the tuning step to find better rates and achieve high-quality translations. In this study, minimum error rate training (MERT) is used to tune the systems [64] [16]. MERT runs the

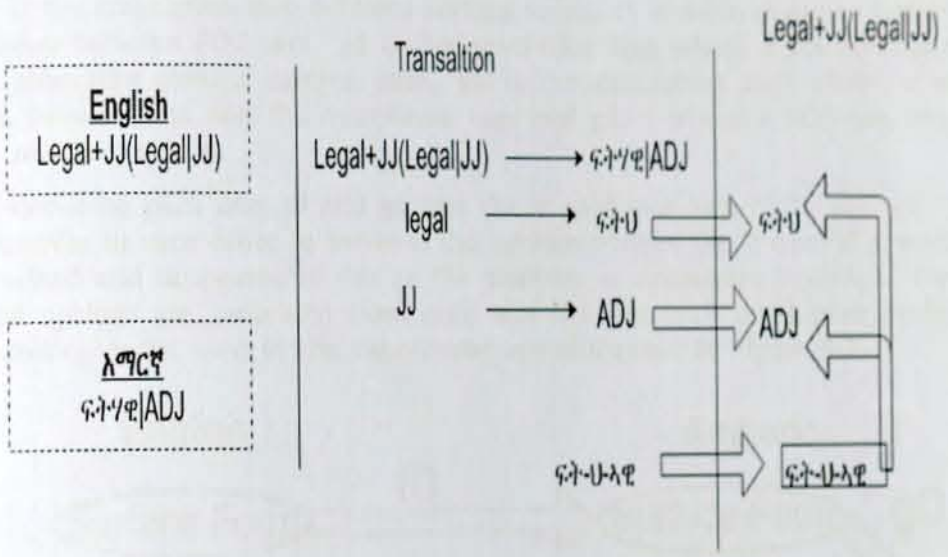


FIGURE 5.4: Generation factors

Moses decoder on the same tuning data several times and finds the best set of weights that provide the best quality translations in terms of BLEU scores.

In this study, the number of maximum iterations of MERT is limited to 10 for all systems because tuning is the slowest step of the process. The tuning of factored systems is similar to the tuning of the phrase-based systems. The configuration file contains input and output factors and the system is tuned according to these factors. In the baseline phrase-based systems, the input and output factors in the configuration file are both 0, which means there is only one factor used.

## 5.2 Decoding

Decoding is simply creating the translation hypothesis from the input text. Using the translation and language models, the decoder machine, namely the Moses toolkit decoder, looks for the best translation of an input text. Moses uses a beam search algorithm which allows keeping all the hypotheses in stack data structures according to their translated word counts. Each time a hypothesis is placed into a stack, the stack may need pruning which means hypotheses with lower scores are removed and then new hypotheses are created from that hypothesis. This is done in iterations until all the input words are translated into the target language.

### 5.2.1 Factored Decoding

Factored systems allow us to have multiple decoding paths. In this study, we use two decoding paths:

1. t0,g0
2. t1,g1,t2,g2

where t0 is the translation step between surface forms, t1 is the translation between lemmas, t2 is translation between POS tags. g0 is the generation step where a lemma and the morpheme tags are generated from a surface form, g1 is the generation step where a surface form is generated from lemma and the morpheme tags and g2 is where a POS tag form is generated from the morpheme tags.

The first decoding path uses t0 and g0 and the second one uses t1,t2, and g1. The two paths are alternatives to each other to increase the performance of the system if a word has been left un-lemmatized and unsegmented due to the analyzer or segmenter coverage. During decoding, translation options are generated from each and the one with the higher probability is used. These decoding paths used in the experiment are illustrated in Figure 5.5.

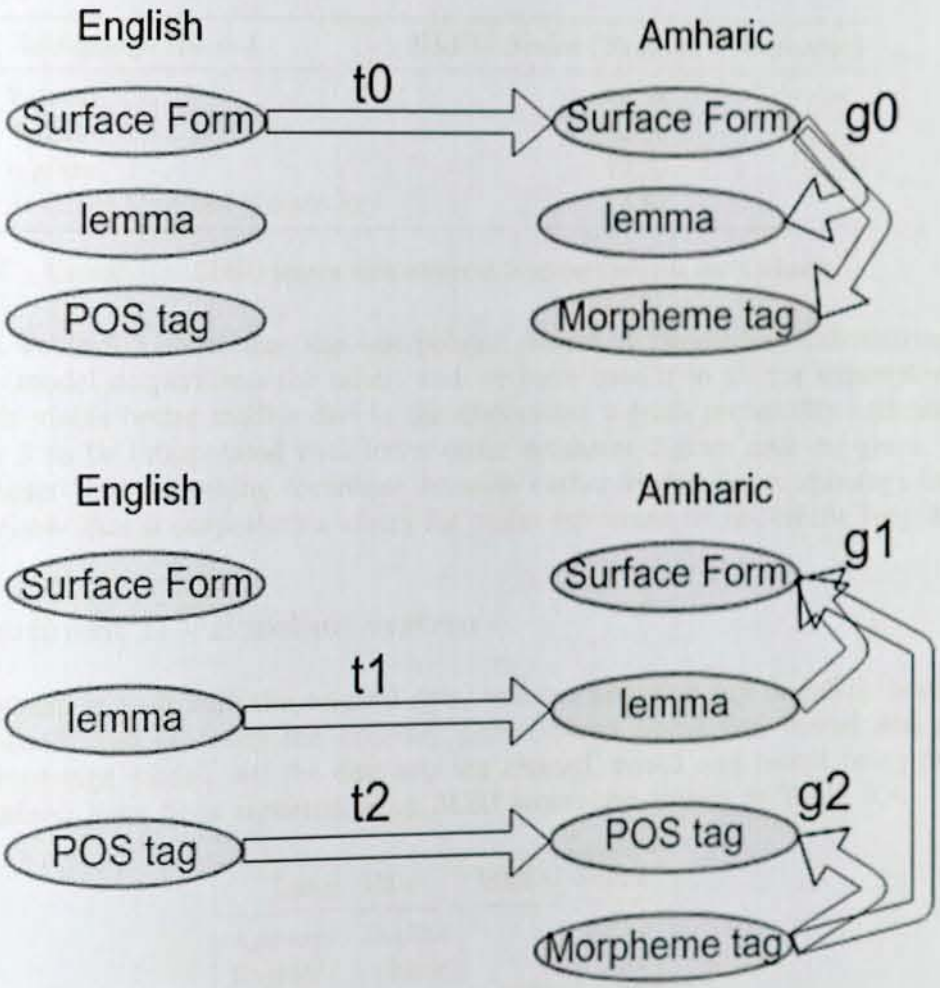


FIGURE 5.5: Decoding paths for the factored system

## 5.3 Experiment Results

### 5.3.1 Experiment I - N-Gram Language Model Effect Experiments

Since we were developing three different language models for our experiments, we first explored the best language modeling N-gram order to use for all the experiments. Considering we have a very limited monolingual data as similar as the bilingual text, we produce 2-gram, 3-gram, and 4-gram language models and performed experiments with the test data on baseline system with each language model for both languages to see the effect on translation fluency. The BLEU results of each language model experiment are listed in Table 5.3.

| Language Model | BLEU Score (English - Amharic) |
|---|---|
| 2-gram | 12.56 |
| 3-gram | 13.04 |
| 4-gram | 12.20 |
| 3- Gram Modified Kneser-Ney | 13.80 |

TABLE 5.3: BLEU scores with different language models for Amharic

The results on Table 5.3 show that the interpolated Modified Kneser-Ney Discounting [63] 3-gram language model outperforms the others and we have used it in all the experiments inside this study. This yields better models due to the discounted 3-gram probability estimates at the specified order 3 to be interpolated with lower-order estimates 2-gram and uni-gram. we have chosen this Kneser-Ney smoothing technique because earlier studies in morphology-based language models show that it outperforms others for under-resourced Ethio-Semitic languages [34].

### 5.3.2 Experiment II - Baseline System

The baseline system is built with the original data, without applying any linguistic feature. This experiment is conducted by using the ordinary phrase-based translation model and a 3-gram surface-based language model. All the data-sets are trained, tuned and tested using Moses and the results obtained have been reported using BLEU scores are shown in Table 5.4.

| Lang. Pair | BLEU Score |
|---|---|
| Amharic - English | 23.12 |
| English - Amharic | 9.52 |

TABLE 5.4: BLEU scores for Baseline Experiment Results

The results show that translating to English has a relatively better score accuracy of 23.12 than translating to Amharic, a BLEU score of 9.52 which lower by 58% in performance to its vice-versa translation. This is since the Amharic corpus has almost half a lower vocabularies than the English side. This makes sparsity in the alignment for Amharic words to match their exact English translation while creating a phrase-table which even could not exist due to morphology inflection on the English counterpart.

### 5.3.3 Experiment III - Surface and POS tag System

Out of all the POS tagging Experiments conducted we choose our best model and other two randomly selected taggers to see the effect of POS tagger accuracy on the translation Accuracy. And it turns out to be, POS tagger accuracy and translation accuracy are directly proportional in translation systems where POS tag is incorporated bidirectionally for both Amharic and English. Results are shown in Table 5.5.

| Lang. Pair | POS tagger | Tagger Accuracy | BLEU Score |
|---|---|---|---|
| Amharic - English | TnT | 80.46 | 22.54 |
| « | CRFs | 80.01 | 21.98 |
| « | Naive-Bays Backoffed* | 90.2 | **24.33** |
| English - Amharic | TnT | 80.46 | 9.24 |
| « | CRFs | 80.01 | 9.58 |
| « | Naive-Bays Backoffed* | 90.2 | **13.80** |

TABLE 5.5: BLEU scores POS tagged system

This method of adding a POS tag into a translation has improved the translation accuracy of both translating to Amharic and English. Much of its improvement is noted on the Amharic side where the POS tag translation has improved the translation by **31%** from a BLEU score of **9.52** to **13.80**.

These improvements have been achieved by preferring a sequence of POS tags where words have not been seen next to each other before using the separate POS tag-set language model and direct POS to POS translation, i.e "JJ" in English is "ADJ" in WIC Amharic Corpus. On the other side, the Amharic POS tag-set language model will tell the translation that a sentence is more likely to start by a Noun(N) and an adjective(ADJ) is most like to be followed by a noun.

### 5.3.4 Experiment III - Surface with Lemma Experiments

In order to see the effect of morphology analysis and segmentation ,we have incorporated one further linguistic information to the surface word. we have added lemma for both Amharic and English using lemmatizers discussed in Chapter 4 Section 4.5. Incorporating lemma have increased the translation accuracy further for both Amharic and English. The BLEU score improvements due to the incorporation of the lemma of both Amharic and English words are as shown in Table 5.7.

| Lang. Pair | BLEU Score |
|---|---|
| Amharic - English | 23.54 |
| English - Amharic | 12.98 |

TABLE 5.6: BLEU scores for Surface + Lemma + POS tagged Experiments

To see the effect of morphology analysis and segmentation ,we have incorporated one further linguistic information to the surface word. we have added lemma for both Amharic and English using lemmatizers discussed in Chapter 4 Section 4.5. Incorporating lemma have increased

the translation accuracy further for both Amharic and English. The BLEU score improvements due to the incorporation of the lemma of both Amharic and English words are as shown in Table 5.7.

### 5.3.5 Experiment IV - Surface with Lemma and POS Experiments

Adopting the best POS tagging model from Experiment III, we have incorporated one further linguistic information to the Experiment III so that we could use both POS tag and lemma with the surface word. This Experiment is conducted by combining the features in Experiment II and III. we have added lemma for both Amharic and English using lemmatizers discussed in Chapter 4 Section 4.5. Incorporating lemma have increased the translation accuracy further for both Amharic and English. The improvements are as shown in Table 5.7.

| Lang. Pair | BLEU Score |
|---|---|
| Amharic - English | 25.48 |
| English - Amharic | 15.28 |

TABLE 5.7: BLEU scores for Surface + Lemma + POS tagged Experiments

Since lemmas are the smallest linguistic features that hold meanings especially in Amharic, challenges in translation due to the morphological richness of Amharic has been further minimized by adding lemma as feature factor. As it can be noted in table both Amharic and English have profited, especially translating to Amharic has shown an increase in BLEU score from 13.80 to 15.28, a 9.6% increase from the last best model in Experiment II and 37.69% improvement from the baseline.

### 5.3.6 Experiment V - Surface with Lemma, POS and Morpheme Segmentation

Considering Amharic is morphologically very rich, by incorporating another Morpheme segmentation information as a feature, to Experiment III for the Amharic side we have found improvements of a 0.56 BLEU Score. Compared to the other linguistic feature in the above experiments morpheme segments has improved the translation accuracy limited. It is only a 3% difference on translation accuracy as shown in the Table 5.8.

| Lang. Pair | BLEU Score |
|---|---|
| Amharic - English | 25.48 |
| English - Amharic | 15.84 |

TABLE 5.8: BLEU scores Surface + Lemma + POS tagged + Morpheme's experiments

## 5.4 Discussion

The results in all the above experiments show that incorporating linguistic features on Amharic data had an increment of 5.96 points in BLEU score a 39.9% translation improvement over the

baseline system while translating to Amharic. Similarly incorporating POS tag and lemma has profited improvement of 2.36 in BLEU score, a 9.26% translation improvement. The comparative BLEU scores of all all the experiments incorporating linguistic features against a baseline system is shown in Figure 5.6.

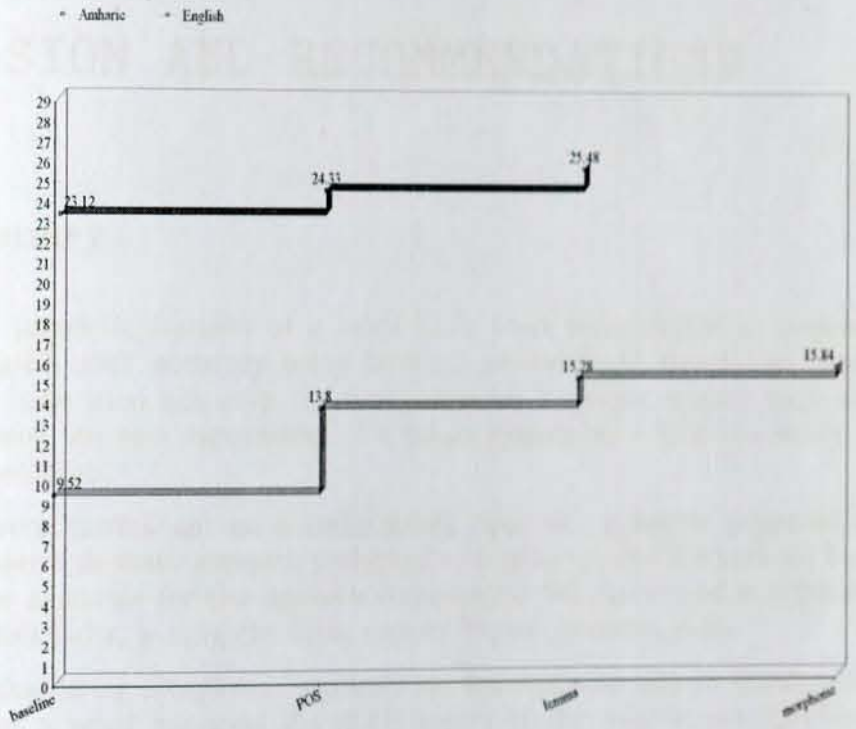


FIGURE 5.6: BLEU Score for all baseline to factored Amharic- English Machine Translation

In the factored models, we use the morphologically segmented representation of Amharic words as a third feature along with lemma, POS tag and the token itself. When we investigate further into this, we see that the tagged systems also perform much better than the baseline. We recommend the incorporation of all possible language features as long as they exist in the nature of language, since the incorporation of one more linguistic feature shows the improvement of the translation model.

We have seen adding morpheme segments has little effects in translation accuracy than adding lemma of a word for Amharic. This is due to the cases that it creates a mismatch with the English version of the data as the English version does not have such a linguistic feature due to its low morphological inflection.

In order to obtain better results with factored models, an additional experiment can be done with adding extra decoding step make use of the already known translations of phrase to phrase indications in the baseline. These combinations would be computing resource intensive in a CPU environment. Moreover, feature reduction segmentation of morphological tags may also improve the results. These methods are introduced in agglutinative Turkish language to English translation and have been applied successfully in some studies [11].

# Chapter 6

# CONCLUSION AND RECOMMENDATIONS

## 6.1 summary

In this study, linguistic features of a word have been incorporated to improve bidirectional Amharic - English SMT accuracy using factored phrase-based translation models. With this approach, we have used not only the most common linguistic feature such as POS tags and lemma of a word but also morphemes of a token employing a factored model to improve the quality of translations.

Experiments were carried out on a most recent data set, a corpus prepared at Addis Ababa University, under a thematic research project of collecting parallel Corpora for bi-lingual English-Ethiopian pairs available for the research community. We have used a segment of the corpus Amharic -English pairs, a religious bible text of 30,646 sentence pairs.

Results show that using morpheme segments on the Amharic side in combination with lemma and POS tag of a word improves the BLEU scores of the translations significantly. The best results we have using these factored phrase-based model obtained with the same data used for an ordinary baseline system, increasing the BLEU score from 9.52 to 15.84 form English to Amharic translation. We also improve the accuracy up to a 25.48 BLEU score from 23.12 for Amharic to English translation.

To summarize, this study introduces an approach of applying morpheme segments on the Amharic data and lemmatization on the English data to build an English to Amharic statistical machine translation system. The results of this study are compared to the benchmark systems which were built with the same data sets and are found to be significantly higher than those.

## 6.2 Conclusion

Experimental results have proven linguistic information incorporation for Amharic -English translation can improve a translation significantly as it did for Turkish, Czech, Tamil and Arabic. The results have had a very similar improvement rate with this morphologically rich languages, implying with the proper POS tagging and morphological analyzer this enriched factored translations model has significant merits, disproving the decrement results reported for English - Tigrigna languages.

On the other hand, out of all the linguistic features incorporated for Amharic - English translation each of them has a different improvement over the translation. For Amharic in terms of BLEU score, POS tag has the most improved score of 4.28, lemma of a word has 1.48, and morpheme

segmentation has 0.56. But importantly the more features used together has the more translation improvement which has been noted to have similar adding all the improvements by each of them.

In this study, our major challenge was computational resources where we incorporating three different features which means three times the data size of the tokens, which restrict us to use limited tuning iterations. Besides, the low computing problems we have spent much time on POS tagger development, morphology analyzer processing tasks; rather than on the main experiments of incorporating linguistic features. On the other hand lack of syntactic parser for Amharic has limited us from conducting hierarchical translation models with linguistic features. Late 2018 early reports in MT have reported translation system based on deep leaning can learn linguistic features implicitly without directly specifying them which is another endeavor to explore.

## 6.3 Recommendation

Based on the finding of this study, the challenges faced and literature in the field, the following recommendations are forwarded.

- The techniques used and suggested throughout this study can be applied for other Ethio-Semitic languages that have a similar nature of inflectional and derivational morphology, especially for Tigrigna and Ge'ez. We have a strong belief that this study can form a basis for future research in the field for these languages.
- A more specific POS tag-set for Amharic could improve a translation since POS has a significant effect on translation accuracy.
- As long as a feature exists in the language and added in a uniform fashion, integrating all the possible features together has a positive impact on the phrase-based factored SMT in Amharic - English Translation.

# References


[1] O Bojar, "English-to-Czech factored machine translation", in *ACL*, 2007, pp. 232–239. DOI: 10.3115/1626355.1626390. [Online]. Available: http://acl.ldc.upenn.edu/W/W07/W07-0735.pdf.

[2] M. Anand Kumar, V. Dhanalakshmi, K. P. Soman, and S. Rajendran, "Factored statistical machine translation system for English to Tamil language", *Pertanika Journal of Social Science and Humanities*, vol. 22, no. 4, pp. 1045–1061, 2014, ISSN: 22318534.

[3] D. Arnold and R. L. Humphreys, *Machine Translation An Introductory Guide*. 1994.

[4] P. Koehn and H. Hoang, "Factored Translation Models", *Computational Linguistics*, no. June, pp. 868–876, 2007. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1091.

[5] T. Gebreegziabher and L. Besacier, "Preliminary Experiments on English-Amharic Statistical Machine Translation", vol. 2, no. 1, pp. 1–6, 2011.

[6] A. Tadesse and Y. Mekuria, "English to Amharic Machine Translation Using SMT", *The Prague Bulletin of Mathematical Linguistics*, no. July, pp. 1–10, 2012.

[7] Amharic.com, *Amharic.com - Learn to Speak, Read & Write in Amharic*, 1996. [Online]. Available: http://amharic.com (visited on 02/15/2018).

[8] B. Gambäck, F. Olsson, A. A. Argaw, and L. Asker, "Methods for Amharic part-of-speech tagging", *Proceedings of the First Workshop on Language Technologies for African Languages - AfLaT '09*, no. March, p. 104, 2009. DOI: 10.3115/1564508.1564527. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1564508.1564527.

[9] E. T. Advisor, Y. Assabie, and M. Gebreegziabher, "Bidirectional English-Amharic Machine Translation: An Experiment using Constrained Corpus", Addis Ababa University, Tech. Rep. March, 2013, pp. 1–109.

[10] T. Tariku, "ENGLISH -TIGRIGNA FACTORED STATISTICAL MACHINE TRANSLATION", ADDIS ABABA UNIVERSITY, Tech. Rep. June, 2014, pp. 1–84.

[11] R. Yeniterzi and K. Oflazer, "Syntax-to-Morphology Mapping in Factored Phrase-Based Statistical Machine Translation from English to Turkish", in *the 48th annual meeting of the ACL*, 2010, pp. 454–464, ISBN: 9781617388088.

[12] I. Youssef, M. Sakr, and M. Kouta, "Linguistic Factors in Statistical Machine Translation Involving Arabic Language", vol. 9, no. 11, pp. 154–159, 2009.

[13] R. Zbib and A. Soudi, *Challenges for Arabic machine translation*. Amsterdam / Philadelphia: John Benjamins Publishing Company, 2012, p. 166.

[14] J. Hutchins, "Multiple Uses of Machine Translation and Computerised Translation Tools", *Proceedings of the International Symposium on Data and Sense Mining, Machine Translation and Controlled Languages - ISMTCL 2009*, pp. 13–20, 2009. [Online]. Available: http://www.hutchinsweb.me.uk/Besancon-2009.pdf.

[15] D. Jurafsky and J. H. Matin, *Speech and Language Processing*. 2008, p. 1038, ISBN: 0130950696.

[16] P. Koehn, *Statistical Machine Translation*. Cambridge: CAMBRIDGE UNIVERSITY PRESS, 2010, ISBN: 9780521874151. [Online]. Available: www.cambridge.org/9780521874151.

[17] T. Poibeau, *Machine Translation The MIT Press Essential Knowledge Series*. London: The MIT Press, 2017, p. 209, ISBN: 9780262534215.

[18] P. Koehn, H Hoang, and A. Birch, "Moses: Open source toolkit for statistical machine translation", *Proceedings of the 45th ...*, no. June, pp. 177–180, 2007. DOI: 10.3115/1557769.1557821. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557821.

[19] C. Goutte, N. Cancedda, M. Dymetman, and G. Foster, *Learning Machine Translation*. 2008, ISBN: 9780262072977. DOI: 10.7551/mitpress/9780262072977.001.0001. [Online]. Available: http://mitpress.universitypressscholarship.com/view/10.7551/mitpress/9780262072977.001.0001/upso-9780262072977.

[20] Anand Ballabh and D. U. C. Jaiswal, "A STUDY OF MACHINE TRANSLATION METHODS AND THEIR CHALLENGES", *International Journal of Advance Research In Science And Engineering*, vol. 8354, no. 4, pp. 1–4, 2015. [Online]. Available: http://www.ijarse.com.

[21] M. D. Okpor, "Machine Translation Approaches: Issues and Challenges", *International Journal of Computer Science Issues*, vol. 11, no. 5, pp. 159–165, 2014.

[22] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation : Parameter Estimation", *Computational Linguistics*, 1993, ISSN: 08912017. DOI: 10.1080/08839514.2011.559906.

[23] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, P. S. Roossin, T. J. Watson, S Della Pietra, V Della Pietra, F. Jelinek, J. D. Lafferty, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "Statistical approach to machine translation", *English*, 1990, ISSN: 08912017. DOI: 10.3115991365.991407.

[24] S. F. Chen and J. Goodman, "Empirical study of smoothing techniques for language modeling", *Computer Speech and Language*, 1999, ISSN: 08852308. DOI: 10.1006/csla.1999.0128. arXiv: 9606011 [cmp-lg].

[25] K. Yamada and K. Knight, "A syntax-based statistical translation model", in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*, 2001. DOI: 10.3115/1073012.1073079.

[26] D. Chiang, "A hierarchical phrase-based model for statistical machine translation", in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, 2005, ISBN: 1932432515. DOI: 10.3115/1219840.1219873.

[27] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: An advanced freely available hierarchical machine translation toolkit", *Machine Translation*, vol. 26, no. 3, pp. 197–216, 2012, ISSN: 09226567. DOI: 10.1007/s10590-011-9120-y.

[28] Y. T. Martha, *Machine Translation lecture notes*, 2017.

[29] M. Hailegebreal, "Bidirectional Tigrigna – English Statistical Machine Translation", Addis Ababa University, Tech. Rep., 2017.

[30] M. Anand Kumar, V. Dhanalakshmi, K. P. Soman, and S. Rajendran, "Factored Statistical Machine Translation System for English to Tamil Language", *Pertanika Journals of SOCIAL SCIENCES & HUMANITIES*, vol. 22, no. 4, pp. 1045–1061, 2014.

[31] H. IMREN, "Head Finalization and Morphological Analysis in Factored Phrase-Based Statistical Machine Translation from English to Turkish", Middle East Technical University, Tech. Rep., 2015.

[32] Baye Yimam, *Amharic Grammer*, 3rd. Addis Ababa: Addis Ababa Univesity Press, 2016, p. 487.

[33] M. Gasser, *HLW: Appendices: Languages Cited*, 2011. [Online]. Available: http://www.indiana.edu/{~}hlw/Appendices/languages.html (visited on 02/15/2018).

[34] Y. T. Martha, "Morphology-Based Language Modeling for Amharic", Universität Hamburg, Tech. Rep. August, 2010.

[35] M. G. Wondwossen, "Machine Learning of Complex Morphology : the Case of Amharic Verbs", Addis Ababa University as, Tech. Rep. July, 2015, p. 288.

[36] Y. Firdyiwek and D. Yaqob, "The System for Ethiopic Representation in ASCII", 1997.

[37] B. G. Gebre, "Part of speech tagging for Amharic", UNIVERSITY OF WOLVERHAMPTON, Tech. Rep. June, 2010.

[38] S. F. Adafre, "Part of Speech tagging for Amharic using Conditional Random Fields", *Computational Linguistics*, 2005.

[39] A. G. Avanzati and Beatriz, *Amharic*, 2013. [Online]. Available: `http://www.languagesgulper.com/eng/Amharic.html` (visited on 12/28/2018).

[40] A. Mulusew, *The Syntax of Non-verbal Predication in Amharic and Geez*. 2014, ISBN: 9789460931543.

[41] C. H. DAWKINS, *The Fundamentals of Amharic*. 1969.

[42] C. W Isenberg, *Grammar of the Amharic Language*, 1st Editio. London: Ann Arbor, 1984.

[43] G. D. Little, "WORD ORDER FUNCTION TYPOLOGY: THE AMHARIC CONNECTION Greta", *Studies in African Linguistics*, 1977.

[44] S. Teferra Abate, M. M. Woldeyohannis, M. Y. Tachbelie, M. Meshesha, S. Atinafu, W. Mulugeta, Y. Assabie, H. Abera, B. Ephrem, T. Abebe, W. Tsegaye, A. Lemma, T. Andargie, and S. Shifaw, "Parallel Corpora for bi-lingual English-Ethiopian Languages Statistical Machine Translation", in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3102–3111. [Online]. Available: `https://www.researchgate.net/project/Parallel-Corpora-for-bi-lingual-English-Ethiopian-Languages-Statistical-Machine-Translation`.

[45] S. T. Abate, M. Y. Tachbelie, S. Atinafu, Y. Assabie, B. Ephrem, W. Tsegaye, T. Andargie, S. Shifaw, A. Lemma, T. Abebe, H. Abera, W. Mulugeta, M. Meshesha, and M. M. Woldeyohannis, "Parallel Corpora for bi-Directional Statistical Machine Translation for Seven Ethiopian Language Pairs", in *October*, vol. 0, 2018, pp. 153–156, ISBN: 9781424436798.

[46] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research", *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, ISSN: 02767783. DOI: 10.2307/25148625. arXiv: `/dl.acm.org/citation.cfm?id=2017212.2017217 [http:]`. [Online]. Available: `http://dblp.uni-trier.de/rec/bibtex/journals/misq/HevnerMPR04`.

[47] G. G. A. Demeke and M. Getachew, "Manual annotation of Amharic news items with part-of-speech tags and its challenges", *Ethiopian Languages Research Center Working Papers*, vol. 2, no. 1, pp. 1–16, 2006. [Online]. Available: `http://nlp.amharic.org/research/papers/by-year/2006/tagging-girmaandmesfin.pdf`.

[48] P. Koehn, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, and C. Moran, "Moses", *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions - ACL '07*, no. June, p. 177, 2007. DOI: 10.3115/1557769.1557821. [Online]. Available: `http://portal.acm.org/citation.cfm?doid=1557769.1557821`.

[49] M. Gasser, "HornMorpho 2.5 User 's Guide", Indiana University, Tech. Rep., 2012, pp. 1–55. [Online]. Available: `http://www.cs.indiana.edu/{-}gasser/Research/projects.html`.

[50] S. Bird, S. Bird, and E. Loper, "NLTK : The natural language toolkit NLTK : The Natural Language Toolkit", *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, no. March, pp. 63–70, 2016. DOI: 10.3115/1225403.1225421. arXiv: 0205028 [cs].

[51] E. AI, *Spacy Industrial-Strength Natural Language Processing*, 2018. [Online]. Available: `https://spacy.io/`.

[52] Z. Li, C. Callison-Burch, C. Dyer, J. Ganitkevitch, S. Khudanpur, L. Schwartz, W. N. G. Thornton, J. Weese, and O. F. Zaidan, "Joshua: An Open Source Toolkit for Parsing-based Machine Translation", *Fourth Workshop on Statistical Machine Translation*, no. March, pp. 135–139, 2009. [Online]. Available: `http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:Joshua:+An+Open+Source+Toolkit+for+Parsing-based+Machine+Translation{\#}0`.

[53] S. Green, D. Cer, and C. Manning, "Phrasal: A Toolkit for New Directions in Statistical Machine Translation", *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 114–121, 2014. [Online]. Available: http://www.aclweb.org/anthology/W/W14/W14-3311.

[54] J. Giménez and L. Màrquez, "Asiya : An Open Toolkit for Automatic Machine Translation ( Meta- ) Evaluation Jesús Giménez , Lluís Màrquez", *The Prague Bulletin of Mathematical Linguistics*, no. 94, pp. 77–86, 2010, ISSN: 1804-0462. DOI: 10.2478/v10108-010-0022-6. PBML. [Online]. Available: http://ufal.mff.cuni.cz/pbml/94/art-gimenez-marques-evaluation.pdf{\%}5Cnhttp://ufal.mff.cuni.cz/pbml/94.

[55] Q. Gao and S. Vogel, "Parallel Implementations of Word Alignment Tool", *SETQA-NLP '08 Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 2008. DOI: 10.3115/1622110.1622119.

[56] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: An open source toolkit for handling large scale language models", in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008, pp. 1618–1621.

[57] *Python*. [Online]. Available: https://www.python.org/ (visited on 06/06/2018).

[58] K. Papineni, S. Roukos, T. Ward, and W. W.-j. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation", *ACL*, 2002, ISSN: 00134686. DOI: 10.3115/1073083.1073135. arXiv: 1702.00764.

[59] G. A. Miller, "WordNet: a lexical database for English", *Communications of the ACM*, 1995, ISSN: 00010782. DOI: 10.1145/219717.219748.

[60] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: The 90% Solution", *NAACL*, 2006.

[61] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: the penn treebank", *Computational Linguistics*, 1993, ISSN: 08912017. DOI: 10.1162/coli.2010.36.1.36100.

[62] Y. T. Martha, T. A. Solomon, and L. Besacier, "Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages – The Case of Amharic", in *Conference on Human Language Technology for Development*, 2011.

[63] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable Modified Kneser-Ney Language Model Estimation", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, ISSN: 0002-9122.

[64] F. J. Och, "Minimum error rate training in statistical machine translation", in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, 2003, ISBN: 9781905593446. DOI: 10.3115/1075096.1075117.

# Appendix A

# The Ethiopic script in ASCII (adoped from Yakob, Firdyiwek [36])

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | (12) | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ግዕዝ | ካዕብ | ሳልስ | ራብዕ | ሃምስ | ሳድስ | ሳብዕ | ዲቃላ → | | | | | |
| ሀ | *he* | *hu* | *hi* | *ha* | *hE* | *h* | *ho* | | | | | | |
| ለ | *le* | *lu* | *li* | *la* | *lE* | *l* | *lo* | | | | *lWa* | | |
| ሐ | *He* | *Hu* | *Hi* | *Ha* | *HE* | *H* | *Ho* | | | | *HWa* | | |
| መ | *me* | *mu* | *mi* | *ma* | *mE* | *m* | *mo* | *mWe* | *(mWu)* | *mWi* | *mWa* | *mWE* | *mW* |
| ሠ | *'se* | *'su* | *'si* | *'sa* | *'sE* | *'s* | *'so* | | | | *'sWa* | | |
| ረ | *re* | *ru* | *ri* | *ra* | *rE* | *r* | *ro* | | | | *rWa* | | |
| ሰ | *se* | *su* | *si* | *sa* | *sE* | *s* | *so* | | | | *sWa* | | |
| ሸ | *xe* | *xu* | *xi* | *xa* | *xE* | *x* | *xo* | | | | *xWa* | | |
| ቀ | *qe* | *qu* | *qi* | *qa* | *qE* | *q* | *qo* | *qWe* | *(qWu)* | *qWi* | *qWa* | *qWE* | *qW* |
| ቐ | *'qe* | *'qu* | *'qi* | *'qa* | *'qE* | *'q* | *'qo* | | | | | | |
| ቀ | *Qe* | *Qu* | *Qi* | *Qa* | *QE* | *Q* | *Qo* | *QWe* | *(QWu)* | *QWi* | *QWa* | *QWE* | *QW* |
| በ | *be* | *bu* | *bi* | *ba* | *bE* | *b* | *bo* | *bWe* | *(bWu)* | *bWi* | *bWa* | *bWE* | *bW* |
| ቨ | *ve* | *vu* | *vi* | *va* | *vE* | *v* | *vo* | | | | *vWa* | | |
| ተ | *te* | *tu* | *ti* | *ta* | *tE* | *t* | *to* | | | | *tWa* | | |
| ቸ | *ce* | *cu* | *ci* | *ca* | *cE* | *c* | *co* | | | | *cWa* | | |
| ኀ | *'he* | *'hu* | *'hi* | *'ha* | *'hE* | *'h* | *'ho* | *hWe* | *(hWu)* | *hWi* | *hWa* | *hWE* | *hW* |
| ነ | *ne* | *nu* | *ni* | *na* | *nE* | *n* | *no* | | | | *nWa* | | |
| ኘ | *Ne* | *Nu* | *Ni* | *Na* | *NE* | *N* | *No* | | | | *NWa* | | |
| አ | *e/a˜* | *u/U* | *i* | *A/a* | *E* | *I* | *o/O* | *ea* | | | | | |
| ከ | *ke* | *ku* | *ki* | *ka* | *kE* | *k* | *ko* | *kWe* | *(kWu)* | *kWi* | *kWa* | *kWE* | *kW* |
| ኸ | *'ke* | *'ku* | *'ki* | *'ka* | *'kE* | *'k* | *'ko* | | | | | | |
| ኸ | *Ke* | *Ku* | *Ki* | *Ka* | *KE* | *K* | *Ko* | *KWe* | *(KWu)* | *KWi* | *KWa* | *KWE* | *KW* |
| ኻ | *Xe* | *Xu* | *Xi* | *Xa* | *XE* | *X* | *Xo* | | | | | | |
| ወ | *we* | *wu* | *wi* | *wa* | *wE* | *w* | *wo* | | | | | | |
| ዐ | *'e* | *'u/'U* | *'i* | *'A/'a* | *'E* | *'I* | *'o/'O* | | | | *zWa* | | |
| ዘ | *ze* | *zu* | *zi* | *za* | *zE* | *z* | *zo* | | | | *ZWa* | | |
| ዠ | *Ze* | *Zu* | *Zi* | *Za* | *ZE* | *Z* | *Zo* | | | | *yWa* | | |
| የ | *ye* | *yu* | *yi* | *ya* | *yE* | *y* | *yo* | | | | *dWa* | | |
| ደ | *de* | *du* | *di* | *da* | *dE* | *d* | *do* | | | | *DWa* | | |
| ዸ | *De* | *Du* | *Di* | *Da* | *DE* | *D* | *Do* | | | | *jWa* | | |
| ጀ | *je* | *ju* | *ji* | *ja* | *jE* | *j* | *jo* | | | | | | |
| ገ | *ge* | *gu* | *gi* | *ga* | *gE* | *g* | *go* | *gWe* | *(gWu)* | *gWi* | *gWa* | *gWE* | *gW* |
| ጎ | *'ge* | *'gu* | *'gi* | *'ga* | *'gE* | *'g* | *'go* | | | | | | |
| ጐ | *Ge* | *Gu* | *Gi* | *Ga* | *GE* | *G* | *Go* | *GWe* | *(GWu)* | *GWi* | *GWa* | *GWE* | *GW* |
| ጠ | *Te* | *Tu* | *Ti* | *Ta* | *TE* | *T* | *To* | | | | *TWa* | | |
| ጨ | *Ce* | *Cu* | *Ci* | *Ca* | *CE* | *C* | *Co* | | | | *CWa* | | |
| ጰ | *Pe* | *Pu* | *Pi* | *Pa* | *PE* | *P* | *Po* | | | | *PWa* | | |
| ጸ | *Se* | *Su* | *Si* | *Sa* | *SE* | *S* | *So* | | | | *SWa* | | |
| ጸ | *'Se* | *'Su* | *'Si* | *'Sa* | *'SE* | *'S* | *'So* | | | | | | |
| ፀ | *fe* | *fu* | *fi* | *fa* | *fE* | *f* | *fo* | *fWe* | *(fWu)* | *fWi* | *fWa* | *fWE* | *fW* |
| ፈ | *pe* | *pu* | *pi* | *pa* | *pE* | *p* | *po* | *pWe* | *(pWu)* | *pWi* | *pWa* | *pWE* | *pW* |

FIGURE A.1: The Ethiopic Script in ASCII (adopted from Yaqob, & Firdyiwek [36])

# Appendix B

# Classifier Based POS Tagger using Naive-Bays

LISTING B.1: Pseudo-code od Taggers trained using NLTK

```
import nltk
import pickle
from nltk.tokenize import LineTokenizer
from nltk.corpus.reader import TaggedCorpusReader
from nltk.tag import tnt
from nltk.tag import RegexpTagger
from nltk.tag import AffixTagger

corpus_reader = TaggedCorpusReader('sent-corpus', ['new-train.pos',
'new-test.pos'], sent_tokenizer=LineTokenizer(), sep='|')
corpus = corpus_reader.tagged_sents()
tags = [tag for (word, tag) in corpus_reader.tagged_words()]
defaultTag = nltk.FreqDist(tags).max()
print(defaultTag)

train_reader = TaggedCorpusReader('sent-corpus', ['new-train.pos'],
sent_tokenizer=LineTokenizer(), sep="|")
train_sentence = train_reader.tagged_sents()
test_reader = TaggedCorpusReader('sent-corpus', ['new-test.pos'],
sent_tokenizer=LineTokenizer(), sep="|")
test_sentence = test_reader.tagged_sents()

patterns = [
(r'^\d+$', 'NUMCR'), (r'^\d+.+$', 'NUMOR'), (r'.*\d+$', 'NUMP'),
(rዎች'.*+$', 'N'), (rአች'.*+$', 'N'), (rነው'+$', 'AUX'),
(rእንደሆነ'+$', 'AUX'), (rመሆናቸው'+$', 'AUX'), (rመሆኑ'+$', 'AUX'),
(rየነበረ'+$', 'AUX'), (rሲሆን'+$', 'AUX'), (rሆነ'+$', 'AUX'),
(rቢችልም'+$', 'AUX'), (r'^:+$', 'PUNC'), (r#'^+$', 'PUNC'),
(r'^\?+$', 'PUNC'), (r፧'^+$', 'PUNC'), (r'^::+$', 'PUNC'),
(r::'^+$', 'PUNC'), (r'^\,+$', 'PUNC'), (r'^\-+$', 'PUNC'),
(r፤'^\+$', 'PUNC'), (r፥'^\+$', 'PUNC'), (r:'^\+$', 'PUNC'),
(r፣'^\+$', 'PUNC'), (r:-'^\+$', 'PUNC'), (r*'^\+$', 'PUNC'),
(r'^\;+$', 'PUNC'), (r“'^\+$', 'PUNC'), (r'^\!+$', 'PUNC'),
```

```
(r'^\+$', 'PUNC'), (r'^\+$', 'PUNC'), (r'^\\+$', 'PUNC'),
(r'^//+$', 'PUNC'), (r'^\(+$', 'PUNC'), (r'^\)+$', 'PUNC'),
(r'^\[+$', 'PUNC'), (r'^\]+$', 'PUNC'),
(r—'^\+$', 'PUNC'), (r'.*', 'N')
]

regex_tagger = RegexpTagger(patterns)
pre3_tagger = AffixTagger(train_sentence, affix_length=1,
min_stem_length=2, backoff=regex_tagger)
pre2_tagger = AffixTagger(train_sentence, backoff=pre3_tagger,
affix_length=2, min_stem_length=2)
pre1_tagger = AffixTagger(train_sentence, backoff=pre2_tagger,
affix_length=3, min_stem_length=1,
cutoff=0.001)

suf4_tagger = AffixTagger(train_sentence, backoff=pre1_tagger,
affix_length=-4)

from nltk.tag.sequential import ClassifierBasedPOSTagger

tagger = ClassifierBasedPOSTagger(train=train_sentence,
backoff=suf4_tagger, cutoff_prob=0.665)
print(tagger.evaluate(test_sentence))

with open('models/naive_back_off.pkl', 'wb') as fout:
pickle.dump(tnt_tagger, fout)

print(accuracy)
```

# Appendix C

# Factored Training Sample Shell Program

```
#!/bin/sh

LANG1="en"
LANG2="am"
START_DIR="$HOME/Desktop/data"
EXP_DIR="fact_morph_regenerate_"$LANG2"_"$LANG1

mkdir $HOME/$EXP_DIR
mkdir $HOME/$EXP_DIR/corpus
mkdir $HOME/$EXP_DIR/working
mkdir $HOME/$EXP_DIR/lm
mkdir $HOME/$EXP_DIR/pre_process

SRC="$START_DIR/$LANG1"_"$LANG2/"
SELECTOR_SCRIPS="$START_DIR/scripts/selector_test_tune.py"
COMBINE_FILES="$START_DIR/scripts/combine.py"
MOSSES_TOKEN="$HOME/mosesdecoder/scripts/tokenizer/tokenizer.perl"
MOSSES_TRUE_CASE_MOD="$HOME/mosesdecoder/scripts/recaser/train-truecaser.perl
MOSSES_TRUE_CASE_BULD="$HOME/mosesdecoder/scripts/recaser/truecase.perl"
MOSSES_CLEAN="$HOME/mosesdecoder/scripts/training/clean-corpus-n.perl"
MOSSES_TRN="$HOME/mosesdecoder/scripts/training/train-model.perl"
MOSSES_MERT="$HOME/mosesdecoder/scripts/training/mert-moses.pl"
CORPUS="$HOME/$EXP_DIR/corpus"
PreProcess="$HOME/$EXP_DIR/pre_process"
SPACE_NORM="$START_DIR/scripts/amh_space_normalizor.py"
CHAR_NORM="$START_DIR/scripts/amh_char_normalizor_v2.py"
AMH_TAGGER="$START_DIR/scripts/amharic_tagger_with_model.py"
ENG_TAGGER="$START_DIR/scripts/english_tagger_with_model.py"
EXTRACT_TAGS="$START_DIR/scripts/extract_tag.py"
LM="$HOME/$EXP_DIR/lm"
WORKING="$HOME/$EXP_DIR/working"


cp "$SRC$LANG1.txt"   "$PreProcess/"
cp "$SRC$LANG2.txt"   "$PreProcess/"
```

```
cd $PreProcess
python3  "$SELECTOR_SCRIPS" $LANG1 $LANG2

# # English tokenization and truecasing
"$MOSSES_TOKEN"  -l en < "$PreProcess/train.$LANG1" >
                  "$PreProcess/train.$LANG1.tok.$LANG1"
"$MOSSES_TRUE_CASE_MOD"  --model "$PreProcess/truecase-model.$LANG1"
                  --corpus "$PreProcess/train.$LANG1.tok.$LANG1"
"$MOSSES_TRUE_CASE_BULD"  --model "$PreProcess/truecase-model.$LANG1"
         < "$PreProcess/train.$LANG1.tok.$LANG1"
          > "$PreProcess/train.true.$LANG1"


"$MOSSES_TOKEN" -l en < "$PreProcess/tune.$LANG1"
        > "$PreProcess/tune.$LANG1.tok.$LANG1"
"$MOSSES_TRUE_CASE_MOD"  --model "$PreProcess/truecase-model2.$LANG1"
         --corpus "$PreProcess/tune.$LANG1.tok.$LANG1"
"$MOSSES_TRUE_CASE_BULD"  --model "$PreProcess/truecase-model2.$LANG1"
        < "$PreProcess/tune.$LANG1.tok.$LANG1"
         > "$PreProcess/tune.true.$LANG1"


"$MOSSES_TOKEN" -l en < "$PreProcess/test.$LANG1"
        > "$PreProcess/test.$LANG1.tok.$LANG1"
"$MOSSES_TRUE_CASE_MOD"
        --model "$PreProcess/truecase-model3.$LANG1"
        --corpus "$PreProcess/test.$LANG1.tok.$LANG1"
"$MOSSES_TRUE_CASE_BULD"
        --model "$PreProcess/truecase-model3.$LANG1"
         < "$PreProcess/test.$LANG1.tok.$LANG1"
          > "$PreProcess/test.true.$LANG1"

#Return to
# # Amharic tokenization and character normalaization
cd "$START_DIR/scripts/"

python2 "$SPACE_NORM" "$PreProcess/train."$LANG2
python2 "$SPACE_NORM" "$PreProcess/tune."$LANG2
python2 "$SPACE_NORM" "$PreProcess/test."$LANG2

python2 "$CHAR_NORM" "$PreProcess/train."$LANG2
python2 "$CHAR_NORM" "$PreProcess/tune."$LANG2
python2 "$CHAR_NORM" "$PreProcess/test."$LANG2

mv "$PreProcess/train.$LANG2" "$PreProcess/train.true."$LANG2
mv "$PreProcess/tune.$LANG2" "$PreProcess/tune.true."$LANG2
mv "$PreProcess/test.$LANG2" "$PreProcess/test.true."$LANG2
```

```
"$MOSSES_CLEAN" "$PreProcess/train.true"
        $LANG2 $LANG1 "$PreProcess/train.clean" 1 80

cp "$PreProcess/train.clean."$LANG1
        "$PreProcess/train.clean."$LANG2 "$CORPUS"
cp "$PreProcess/tune.true."$LANG1
        "$PreProcess/tune.true."$LANG2 "$CORPUS"
cp "$PreProcess/test.true."$LANG1
        "$PreProcess/test.true."$LANG2 "$CORPUS"

# # #Tagging with best model yet

cd $CORPUS

python3 "$ENG_TAGGER" "$CORPUS/train.clean.$LANG1"
python3 "$ENG_TAGGER" "$CORPUS/tune.true.$LANG1"
python3 "$AMH_TAGGER" "$CORPUS/train.clean.$LANG2"
python3 "$AMH_TAGGER" "$CORPUS/tune.true.$LANG2"


# # # # language model creation
cd $LM
python3 "$COMBINE_FILES" "$PreProcess/train.true."$LANG1
        "$PreProcess/tune.true."$LANG1 "$LM/combined_lm.$LANG1"
python3 "$COMBINE_FILES" "$PreProcess/train.true."$LANG2
        "$PreProcess/tune.true."$LANG2 "$LM/combined_lm.$LANG2"

python3 "$COMBINE_FILES" "$CORPUS/train.clean_tagged."$LANG1
        $CORPUS/tune.true_tagged."$LANG1 "$LM/combined_tagged_lm.$LANG1"
python3 "$COMBINE_FILES" "$CORPUS/train.clean_tagged."$LANG2
        "$CORPUS/tune.true_tagged."$LANG2 "$LM/combined_tagged_lm.$LANG2"

python3 "$EXTRACT_TAGS" "$LM/combined_tagged_lm.$LANG1" "lemma"
python3 "$EXTRACT_TAGS" "$LM/combined_tagged_lm.$LANG1" "pos"
python3 "$EXTRACT_TAGS" "$LM/combined_tagged_lm.$LANG2" "lemma"
python3 "$EXTRACT_TAGS" "$LM/combined_tagged_lm.$LANG2" "pos"
cd $LM
# ## double the data
python3 "$COMBINE_FILES" "$LM/combined_tagged_lm_lemma.$LANG1"
"$LM/combined_tagged_lm_lemma.$LANG1"
        "$LM/combined_tagged_lm_lemma2.$LANG1"
python3 "$COMBINE_FILES" "$LM/combined_tagged_lm_lemma.$LANG2"  "$LM/combined
        "$LM/combined_tagged_lm_lemma2.$LANG2"
python3 "$COMBINE_FILES" "$LM/combined_tagged_lm_pos.$LANG1"  "$LM/combined_t
        "$LM/combined_tagged_lm_pos2.$LANG1"
python3 "$COMBINE_FILES" "$LM/combined_tagged_lm_pos.$LANG2"  "$LM/combined_t

cd $LM
# ## Language Model
```

```
"$HOME/mosesdecoder/bin/lmplz" -o 3
        --interpolate_unigrams 0 --discount_fallback
        --prune 0 <"$LM/combined_lm.$LANG1" >"$LM/surface.arpa.$LANG1"
"$HOME/mosesdecoder/bin/lmplz" -o 3 --interpolate_unigrams 0
        --discount_fallback
        --prune 0 <"$LM/combined_lm.$LANG2" >"$LM/surface.arpa.$LANG2"

"$HOME/mosesdecoder/bin/lmplz" -o 3
        --interpolate_unigrams 0 --discount_fallback
        --prune 0 <"$LM/combined_tagged_lm_lemma2.$LANG1"
        >"$LM/lemma.arpa.$LANG1"
"$HOME/mosesdecoder/bin/lmplz" -o 3
        --interpolate_unigrams 0
        --discount_fallback --prune 0
        <"$LM/combined_tagged_lm_lemma2.$LANG2" >"$LM/lemma.arpa.$LANG2"

"$HOME/mosesdecoder/bin/lmplz" -o 3 --interpolate_unigrams 0 --discount_fallbac
"$HOME/mosesdecoder/bin/lmplz" -o 3 --interpolate_unigrams 0 --discount_fallbac

# ##Binarization of Language model----------------------------------------------

"$HOME/mosesdecoder/bin/build_binary" "$LM/surface.arpa.$LANG1" "$LM/surface.bl
"$HOME/mosesdecoder/bin/build_binary" "$LM/surface.arpa.$LANG2" "$LM/surface.bl
"$HOME/mosesdecoder/bin/build_binary" "$LM/lemma.arpa.$LANG1" "$LM/lemma.blm.$L
"$HOME/mosesdecoder/bin/build_binary" "$LM/lemma.arpa.$LANG2" "$LM/lemma.blm.$L
"$HOME/mosesdecoder/bin/build_binary" "$LM/pos.arpa.$LANG1" "$LM/pos.blm.$LANG1
"$HOME/mosesdecoder/bin/build_binary" "$LM/pos.arpa.$LANG2" "$LM/pos.blm.$LANG2

# ##Translation Model-----------------------------------------------------------

cd $WORKING

"$MOSSES_TRN" --cores 4 --mgiza --mgiza-cpus 4 --parallel \
--root-dir "$WORKING/train" \
--corpus "$CORPUS/train.clean_tagged" \
--alignment grow-diag-final-and \
--reordering msd-bidirectional-fe \
--f "$LANG2" --e "$LANG1" \
--lm 0:3:"$LM/surface.blm.$LANG1" \
--lm 1:3:"$LM/lemma.blm.$LANG1" \
--lm 2:3:"$LM/pos.blm.$LANG1" \
--translation-factors 3,0-0 \
--decoding-steps t0 \
--external-bin-dir "/home/tsegaye/master-moses-traing-tools/"

cd $WORKING

mkdir "$WORKING/mert-work"
```

```
"$MOSSES_MERT" \
"$CORPUS/tune.true.$LANG2" "$CORPUS/tune.true.$LANG1"\
"$HOME/mosesdecoder/bin/moses" \
"$WORKING/train/model/moses.ini" \
--mertdir "$HOME/mosesdecoder/bin" \
--rootdir "$HOME/mosesdecoder/scripts" \
--batch-mira --return-best-dev \
--decoder-flags '-threads 4'

### ####Binaries----------
cd $WORKING

mkdir "$WORKING/binarised-model"

"$HOME/mosesdecoder/bin/processPhraseTableMin" \
-in "$WORKING/train/model/phrase-table.0-0,1,2.gz" \
-nscores 4 -out "$WORKING/binarised-model/phrase-table.0-0,1,2.minphr"


####### step final
cp "$WORKING/mert-work/moses.ini" "$WORKING/binarised-model"

sed -i -e "s/PhraseDictionaryMemory/PhraseDictionaryCompact/g" "$WORKING/bina
sed -i  -e "s#path=$WORKING/train/model/phrase-table.0-0,1,2.gz#path=$WORKING
## sed -i  -e "s#path=$WORKING/train/model/phrase-table.0-0,2.gz#path=$WORKI

## ####test with file----------

cd $WORKING

"$HOME/mosesdecoder/scripts/training/filter-model-given-input.pl" "$WORKING/f

### ###bleu----------

"$HOME/mosesdecoder/bin/moses" -f "$WORKING/filtered-test."$LANG2"/moses.ini"
## ##run BLEU script----------
"$HOME/mosesdecoder/scripts/generic/multi-bleu.perl" -lc "$CORPUS/test.true."

echo "writting BLEU for EXP-$LANG2-$LANG1 Done!! "
```