School of Computing

Jimma Institute of Technology

Jimma University

Parts of Speech Tagging for Awngi Language

By

Wubetu Barud

A Thesis Submitted to the School of Computing of Jimma University for the Fulfillment of Masters of Science in Information Technology

Jimma, Ethiopia

November, 2016

School of Computing

Jimma Institute of Technology

Jimma University

Parts of Speech Tagging for Awngi Language

Wubetu Barud

Signature of the Board of Examiners for Approval

NameSignatureDate

1. Dr.Melita Luke, Chairperson_____ _____
2. Getachew Mamo, Advisor _____
3. Teferi Kebabaw, Internal Examiner_____ _____

Acknowledgement

First and foremost, I would like to express my heartfelt gratitude to the almighty God. All of my efforts would have gone for nothing if it had not been for his importunate help.

Then I offer my sincerest thanks to my advisers, **Mr. Getachew M**. and **Mrs. Ruth G**., who has supported me throughout my thesis.

Though it is difficult to mention the names of the person who gave me their hand while doing this thesis, it is necessary to mention those who gave me their precious time to pre-processing of the data needed for this thesis, to share ideas, and gave me moral and material support. **Mr. Birhanu Asaye, Mrs. Alemnesh**, **Mr. Tegegn Birhan**, **Mr. Dessalew G**, **Mr. Asaye** and **Mr. Mahitot** are few of them. I am very grateful to thank them for what they did.

It is my pleasure to express my gratitude enthusiastically to my family members. My father, mother, brothers and sisters thank you a lot.

I am also greatly thankful to many of my friends who directly or indirectly supported me in my thesis work, particularly many thanks to **Mr. Gashaw Zemene** and **Mr. Dessalew Getnet**.

Finally, I want to express my sincere gratitude and profound thanks to all my dear friends without whose help this thesis work could not have been completed.

Table of Contents

Contents **Pages**

List of Tables and Figures

Tables

Figures

Acronyms and Abbreviations

NLP	Natural Language Processing
NLPAs	Natural Language Processing Applications
POS	Parts of speech
POST	Parts of speech Tagger
HMM	Hidden Markov Model
MEMM	Maximum Entropy Markov Model
ANN	Artificial Neural Network
TnT	Trigrams'n'Tags
SVM	Support Vector Machine
TBL	Transformation Based Learning
TEL	Transformational Error driven Learning
CRF	Conditional Random Field
MBT/L	Memory Based Tagging/Learning

Abstract

Natural language refers to human languages like Awngi and Amharic as opposed to artificial or programming languages such as C++, Java, Pascal, etc

Natural Language processing (NLP) is the major field of study in computer science and related departments. NLP increase the ability of computersto understand, interpret and communicate using human languages. It is a branch of computational linguistics which is concerned with automated, computer processing of natural language such as speech acts or texts.

Parts of speech tagging, one of the major tasks of NLP, automatically tags the word of a text by labels that can be used to determine the structure of a sentence. The parts of speech tagger developed in this thesis is based on probability theory.

The purpose of this thesis is to develop parts of speech tagger for Awngi language using Hidden Markov Model (HMM). Most natural language processing systems use parts of speech (POS) tagger as one of their components in their system.

Awngi language literatures on grammar and morphology are reviewed to understand nature of the language and also to identify possible tagsets. Based on this, 23 tagsets are identified and for the study, we have collected 350 sentences (with total word of 3760 both for training and testing sets).

The performance of the tagger, Awngi language HMM POS tagger is tested using tenfold cross validation mechanism. The experimental result indicates both unigram and bigram taggers tag words with **85.16**% and **87.84**% accuracy respectively. Based on the achieved result conclusions and recommendations are forwarded

Keywords: NLP; POST; HMM; N-Gram; Awngi Language.

Chapter One

Introduction

1.1. Background

Natural language refers to human languages like Awngi and Amharic as opposed to artificial or programming languages such as C++, Java, Pascal, etc. Natural language processing (NLP) is a computerized approach to analyzing text that is based on both a set of theories and technologies. It also concerns to process and understand natural language using computers. Thus, it performs useful tasks like that of enabling human to machine communication, improving human to human communication, or simply doing useful processing of texts or speeches.

The goal of NLP is to accomplish human-like language processing for various tasks and applications such as machine translation, information retrieval, question-answering, and parts of speech tagging.

Parts of speech tagging is an application of NLP which is the process of labeling words in a text as corresponding to a particular part of speech category. It is usually part of many tasks of NLP applications. It plays an important role in natural language processing applications such as speech recognition, speech synthesis, information retrieval, word sense disambiguation and machine translation [1].

Many indigenous African languages including Awngi are under-resourced that they have very few computational linguistic tools or corpora (such as lexica, taggers, parsers or tree-banks) available. Thus, this study concerns to fill these gaps in developing parts of speech tagger for Awngi language. The Awngi language is a category of Afro-Asiatic family and used in Ethiopia, Sudan and Tanzania [2]. Ge'ez script is used for writing system of the language.

1.2. Statement of the Problem

Parts of Speech Taggers(POST) are mainly to solve ambiguity of languages. As mentioned in section 1.1, POS taggers are mainly to disambiguate ambiguous words based on their context that they are used in sentences. Ambiguous words are common problem in Awngi language. For example,

a. ፋቱኽሳእያታሼውዼስድባኖ**ይሚ**። (To get what you want **ask** God heartily.)
b. ይታላባርዳራካሳውጌርክ**ይሚ**ኑና።(My father will go to BahirDar on **Wednesday**.)
c. ኮቢው**ወይሚ**ውኻዋቱኽ? (What is the **price** of pen?)
d. **ወይሚ**እላቲውዳብቲዳእንጅኩንስ። (let's sit on **grassless** land).

As we have seen in the above four sentences, the words are conveying different meanings based on their sequences. In sentence (a), the word "**ይሚ**" means "**ask**" which is used as a verb, whereas in sentence (b), this word is used as the noun "**Wednesday**". And when we see the third sentence (c), the word "**ወይሚ**" means "**price**" which is used as a noun, whereas in the last sentence (d), the same word is used as adjective "**grassless**" which describes the noun land.

Researches in the area of POS tagging will contribute a lot in the effort of natural language processing of Awngi language. The absence of the POS tagging for Awngi language limits (make difficult) using machine translation, grammar checking, word-sense disambiguation and etc. to understand for the machine (computer), when further study will be held by different researchers.

Due to lack of resources and deficiency of natural language processing tools the study is very necessary to solve those problems. In fact, there are parts of speech taggers developed for local languages like Amharic, Afaan Oromo and Tigrigna etc. But, these part of speech taggers can't be used for other languages [3]. Therefore, the aim of this study is to investigate and develop POS tagger for Awngi language so as to establish the base for future researchers who have an interest in the area of NLP applications. Hence, conducting research and developing an automatic parts of speech tagger (POST) for Awngi language worth paramount significance.

1.3. Objective of the Study

1.3.1. General Objective

The general objective of the study is to investigate the development of parts of speech tagging model for Awngi language.

1.3.2.Specific Objective

To achieve the general objective of the study, the following objectives are specific:

- ✓ To identify and review techniques for POS tagging
- ✓ To study the structure of Awngi sentences
- ✓ To identify word categories and tagsets for Awngi language
- ✓ To prepare sample training corpus for the study
- ✓ To compute lexical and transitional probabilities based on the sample training corpus
- ✓ To test and evaluate the performance of the POS tagging model
- ✓ To present the reports of the study

1.4. Methods and Methodologies

1.4.1.Literature review

We have reviewed documents related to Awngi language to understand the structure of words within a sentence and identify word categories of the language. We have also reviewed literatures related to POS tagging to identify different methods that are used to develop POS taggers and focused on Hidden Markov Model methods.

Referring and analyzing of different documents related to the parts of speech tagging in local and foreign languages helps us an input for this study. Analyzing and reviewing of literatures helps to

understand the models and algorithms that are used to develop for the parts of speech tagging. In this thesis, we reviewed different related literatures that help us to analyze and model the parts of speech tagging for Awngi language. And also uses to understand the grammatical structures of the language as well.

1.4.2.Data sources and preparation Techniques

Data needed for development of parts of speech tagging for Awngi language was collected from different sources in the forms of both hard and soft copies. We have collected 350 (Three Hundred and Fifty) sentences from different sources. Those sources are considered to be under different domains or categories such as teaching materials, Awngi newspaper (which contains news of economic, political, social, and health related aspects). After collecting the important data for the study, pre-processing and identifying the tagsets of the Awngi sentences were done with the help of linguistic experts in the language. After pre-processing, the data are divided into two subsets namely, training and testing data sets and then use tenfold cross validation for experimental purposes.

1.4.3.Hidden Markov Model

For this study, we adopted the Hidden Markov Model method particularly Viterbi algorithm. HMM based tagger relies on the statistical property of words along with their parts of speech categories. Such a statistical property can be distributed probability of words with tags which can be obtained during the training phase of the system. Here, the criteria of selecting the best approach to increase the performance of the application depend on the quality and amount of corpora that we have collected. Similarly, the collected corpus must be balanced and contain different words as well as sentences that includes from different genres.

In order to select the best approach, first we have to see and compare the experimental result of all approaches. For example, let’s see the rule-based approach, it involves manual rule construction which is laborious, it takes many times, prone to error and require deep linguistic knowledge of the language being tagged. But it may contain less information that is stored using the rule and small set of rules, ease of finding and implementing improvements to the tagger. Let’s see the thesis which was done by using Transformational Error driven Learning (TEL)

approach for Afaan Oromo language using 18 tagsets and 223 sentences (1708 words) for experiment [4].

This thesis work scores accuracy for bigram approach is 70.63% and for unigram 68.08% whereas that of original Brill tagger without modification is 77.64 and 80.08% for modified Brill tagger which is less than [1] was done in HMM approach by using 17 tag sets and 159 sentences (1621 words) for experiment and it scores maximum result both in unigram and bigram algorithms even the collected corpora are smaller.

According to[5], the results of their POS tagging experiments for Amharic showed that MBT is a good tagging strategy for under-resourced languages as the accuracy of the tagger is less affected as the amount of training data increases compared with other methods, particularly TnT. Those researchers have used and compared the experimental results by using methodologies like SVM, TnT, CRF and MBT only. But they didn't compare the results by using HMM POS tagger and the work of [6] by using other methodology on other under-resourced and morphologically reach language concluded experimentally with increased performance than the work of [4] which was done in the same language and the POS tagger experimental result is highly affected and improved by increasing the amount of training data. Here, both researchers concluded that, increasing the amount of training data will increase the performance of the tagger.

1.5. Scope and Limitation of the Study

Due to time and budget limitations, we prepared only sample corpus and tagsets at broad level together with linguistic experts for this experimental study. The tagsets used are meant to give information of words about their word class categories only, but not about the issues like gender, number, and tense aspects etc.

1.6. Experimental Analysis

After the parts of speech tagger is developed, it is trained on 90% of the entire collected data and the remaining is used for testing purposes. Accuracy is taken as the performance measure of the model and indicates the closeness of the agreement between the test result and the accepted reference value (the manually tagged text of the test set).

1.7. Application of Results and Beneficiaries

POS tagging is a useful form of linguistic analysis. The application potential of textual corpora increases, when the corpora are annotated. The first logical level of annotation is usually part of speech tagging. Parts of speech (POS) tagging is often considered as the first phase of a more complex natural language processing applications [7]. It is also one of the main tools needed to develop many language corpus [8]. The output of this study has significance to initiate other researchers to participate in different computational researches of Awngi language.

1.8. Organization of the of thesis

The whole thesis is organized into five Chapters. The first Chapter introduces about natural language processing (NLP), parts of speech tagging (POST), a statement of the problem, objectives of the study, methodologies, scope and limitations of the study and applications of the result are included.

The second Chapter is all about literature review and related works. It describes the approaches used so far for POS tagging and works that are done using those approaches specially by using HMM, Hybrid, MEMM approaches have been discussed in detail. Chapter three focuses on study of linguistic properties of Awngi language (the nature, word class, and sentence structure) and tag set preparation of the language are discussed. The fourth Chapter deals with the design and implementation of the Awngi POS tagger including corpus preparation and analyses of the results obtained.

Finally, the last chapter presents conclusions and recommendations about the thesis are included.

Chapter Two

Literature Review and Related work

2.1. Introduction

The main purpose of this chapter is to give an overview of existing literatures and methods used in the field of natural language processing (NLP) applications particularly in parts of speech tagging. Most language processing systems must recognize and interpret the linguistic structures that exist in a sequence of words. This task is virtually impossible if all we know about each word is its text representation. Instead we want to be able to generalize over classes of words. These word classes are commonly named as parts of speech (POS).

As it has been discussed in chapter one, parts of speech tagging (POST), also called grammatical tagging or word class category disambiguation, is the process of labeling or categorizing words in texts with a particular word class, based on both its definition as well as its context i.e. relationship with adjacent and related words within a phrase, sentence or paragraph [9] . It is also a system that assigns the correct parts of speech or word class to each of the words in a document. Classical parts of speech are noun, verb, adjective, adverb and a few others, but nowadays, parts of speech tag sets sub-divided these general word classes into smaller ones, such as noun with conjunction, noun with adjective, noun with adverbs and other sub classes.

There are well-established sets of abbreviations for naming these classes, usually referred to as POS tags (For example, labels such as, NN for Noun, VV for Verbs RB for Adverbs, RBR for Adverb with comparatives and RBS for Adverb with superlatives etc.). There is no standard representation for these parts of speech. Different researchers have used different symbols depending on the number of tag and morphological structure in the language under study. For example, parts of speech tagging guidelines for the Penn Treebank Project [10] uses VB for base form verb, WDT for wh-determiners, JJ for Adjectives, NN for singular proper noun and NNS for plural noun while others uses NN for all nouns, VV for verbs and ADJ for adjectives. But for this thesis work we have used our own sets of abbreviations for naming of those word class

elements. For example, we have abbreviated Adjectives as ADJ, Adverbs as ADV, Verbs as VB and all independent noun groups as NN etc.

The collection of tags used for a particular task is known as a tagset. A corpus is a collection of texts from different areas such as newspaper texts and scientific articles. Corpus in most cases contains extra information about every word such as its parts of speech and morph-syntactic properties [6].

Parts of speech tagsets typically contain many different word classes. It is also a non-trivial task because most words are ambiguous. They can belong to more than one class, the actual class depends on the context of use.

There are many publicly available POS taggers on the web for different foreign languages. For example it is possible to see the English version of Hidden Markov Model (HMM) based parts of speech using Stanford tagger. Example: **We can can the can**. (‘**Can**’ correspond to auxiliary verb, verb and noun respectively). It generates word class information as follows.

Input sentence: **We can can the can**.

And the Output sentence looks like the following:

We/PRP can/MD can/MD the/DT can/NN. /.

Where, PRP=Pronoun, MD=Verb, Modal, NN =noun, singular, common, DT=determinant and. =sentence terminator

And another POS tagger tags this sentence differently. Example, Real Time parts of speech tagger tags it as follows.

Input text= we can can the can.

Output text= we +PRONPERS can +VAUX can +VI the +DET can +NOUN. +SENT

Where, PRONPERS= Personal pronoun, VAUX= Auxiliary verb, VI= Infinitive verb, DET= Determinant, Noun= Noun and SENT= Sentence terminator

Generally, different Parts of speech taggers tag the same word differently.

2.2. Approaches to POS Tagging

There are different approaches to the problem of assigning each word of text with parts of speech tags. Those approaches have been proposed to annotate words automatically with their parts of speech tags from the given corpus. Before implementing parts of speech tagger for the language different approaches and algorithms have to be analyzed and evaluated for their strength and efficiency. The most common ones are rule-based, stochastic, artificial neural network, hybrid, Maximum Entropy Markov Model (MEMM), and Hidden Markov Model (HMM) approaches. The following section describes each of the above approaches.

2.2.1.MEMM (Maximum Entropy Markov Model) Method

Maximum entropy approach is more flexibility with the context, which is used poorly in HMM frame work. It trains from annotated corpus and assigns tags (POS tags and chunk labels) to previously unseen text. This model uses multiple features simultaneously to predict the tag for a word. It is a conditional probabilistic sequence model. It can represent multiple features of a word and can also handle long term dependency. It is based on the principle of maximum entropy which states that the least biased model which considers all known facts are the one which maximizes entropy. Each source state has an exponential model that takes the observation feature as input and output a distribution over possible next state.

The most likely path through the HMM or MEMM would be defined as the one that is most likely to generate the observed sequence of tokens [11][12][13].

An advantage of MEMM rather than HMM for sequence tagging is that they offer increased freedom in choosing features to represent observations. In sequence tagging situations, it is useful to use domain knowledge to design special-purpose features. But it has a drawback of that it potentially suffer from the "label bias problem," where states with low-entropy transition distributions "effectively ignore their observations."

2.2.2.ANN (Artificial Neural Network) Approach

Artificial Neural Network (ANN) is a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. It is generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems. Learning in biological system and ANN involve adjustment of the synaptic connection that exists between neurons.

When ANN approach is taken in to POS tagger developments task, according to [14] before working on the actual ANN based tagger, it requires a pre-processing activity. The output of the pre-processing activity can be taken as input for the input layer of the network. From which the network learns by adapting the weights of the connection between layers until the correct POS tag is produced.

As an input to the neural network takes numerical values encoding of input word into a suitable form, which the network can identify and use is essential. A single neuro tagger takes numerical values as input, which is obtained by encoding the words using prior tag probabilities. The contextual probabilities are left for being learned from the training corpus. Each word “w” from the corpus is encoded as “n” element vector INPUT = (t_1, t_2, t_n) where “n” corresponds to the total number of tags. If word “w” appears in the training data, the vector INPUT comes from the lexicon of words and “N(w)” can be obtained.

Where, N(w) = number of possible POS tags that can be assigned to the word “w”. In order to perform parts of speech tagging by using ANN, first prior tag probabilities of the word, its neighbor’s, root and the value of the length indicator are passed to the input units. After that only forward pass of the error back propagation learning algorithm is allowed and output neuron with largest value is found. The tag corresponding to this output neuron is finally attached to the current word. If the second largest value in the output layer is close to the largest one, the tag

corresponding to the second largest value may be given as an alternative output. So, multiple outputs or a sorted list of all tags as output may be given without any additional computation and the final decision can be delayed to a later processing stage like chunker, parser or a rule based POST processing system can be used to select the most appropriate tag.

2.2.3.Rule Based Approach

Rule based approach uses hand-written rules for tagging. The tagger depends on dictionary or lexicon to get possible tags for each word to be tagged. According to [15], the rules depend on linguistic features of specific language such as morphological, lexical and syntactical information. Hand-written rules are used to identify the correct tag when a word has more than one possible tag. Disambiguation is done by analyzing the linguistic features of the word, its preceding word, its following word and other aspects. Such a like manner of obtaining rule is call Brill Transformation based approach and described in the work of [15] and two types of rules are used.

- ✓ Lexical rules: define the label of the word based on its lexical properties.
- ✓ Contextual rules: define the labeling, that is to say to return to previously assigned labels and correct by examining the local context.

Both types of rules have the following forms:

- ✓ If a word is labeled A is in a context C, then change it to B (contextual rule).
- ✓ If a word has lexical property P, then assign the label a (lexical rule).

2.2.4.Hybrid Approach

Parts of speech taggers sometimes fail to correctly classify cases for which linguists can easily decide the correct parts of speech category. These types of errors are generated due to noise in the training data but also because some linguistic phenomena are not detected by machine learning methods. Hybrid models are basically combination of rule-based and statistical models. This approach uses the combination of both rule-based and machine learning technique and

makes new methods using strongest points from each method. It uses essential features from machine learning approaches and uses the rules to make it more efficient [3][16].

2.2.5.Stochastic Approaches

Most current parts of speech taggers are stochastic. It is also called statistical approach, which is based on a probabilistic pattern to assign a probable parts of speech tags to a given text from a given training text corpus.

The goal of any stochastic approach is to pick the most probable tag for a word from its context and its neighbors [1]. They can build a probability matrix that stores the probability of an individual word belonging to a certain parts of speech category and its distributional probability.

N-gram Method

An N-gram is a contiguous sequence of "N" items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (N − 1) order Markov model.

$$P(\mathrm{w_n}/\mathrm{w_1}^{\mathrm{n-1}}) \approx P(w_{\mathrm{n}}/w_{\mathrm{n-1}})$$

For example, in order to compute a particular bigram probabilities of a word "y" given a previous word "x" , we will compute the count of the bigram that means "count(xy)" and normalize by the sum of all the bigrams that share the same first word x:

$$P(\mathrm{w_n}/\mathrm{w_1}^{\mathrm{n-1}}) = \frac{\mathrm{Count}(\mathrm{w_{n-1}w_n})}{\sum w\ \mathrm{Count}(\mathrm{w_{n-1}w})}$$

After this, by simplifying this formula we can get the following.

$$P(\mathrm{w_n}/\mathrm{w_{n-1}}) = \frac{\mathrm{Count}(\mathrm{w_{n-1}w_n})}{\mathrm{Count}(\mathrm{w_{n-1}})}$$

For example, let's see the following using mini-corpus of four Awngi sentences. We will first need to augment each sentence with a special symbol "<S>" at the beginning of the sentence, to give us a bigram context of the first word and we will also need a special symbol "</S>" at the end of the sentence.

<S>ፋቱኽሳእያታሼውዴስድባኖይሚ</S>

<S>ይታላባርዳራካሳውጌርክይሚኑና</S>

<S>ኮቢውወይሚውኻዋቱኽ</S>

<S>ወይሚእላቲውዳብቲዳእንጅኩንስ</S>

Here we can calculate for some of the bigrams from the above given corpus.

$$P(\text{ፋቱኽሳ}/< s >) = \frac{\text{Count}(\text{ፋቱኽሳ}, < S >)}{\text{Count}(< S >)}$$

Therefore, count ($\text{ፋቱኽሳ}, < S >$) =1 and $\text{Count}(< S >) = 4$

So, $P(\text{ፋቱኽሳ}/< s >) = 1/4$=0.25.

Similarly, $P(</s >/ \text{ይሚ}) = \frac{\text{Count }(\text{ይሚ},</S>)}{\text{Count }(\text{ይሚ})}$= 0.5. By using similar method, we can calculate the probabilities of the other words which are in the mini-corpus. But this method is used to estimate the probabilities of words in the corpus only that means in can't use to calculate probability of tags with the given word and probability of tags with its previous tag in order to maximize the assumptions of HMM POS tagger.

Maximum-Likelihood (Most Frequent Tag) Method

The word likelihood probabilities represent the probability given that we see a given tag, that it will be associated with a given word. It assigns the most frequent parts of speech tag for a token in the training data to a token in untagged data. This can be calculated by counting every word with a specific tag and dividing it with the number of occurrence for this particular tag, which gives conditional probability of the word given the tag. This can be represented mathematically as:

$$p(W/T) = \frac{\text{Count (W,T)}}{\text{Count (T)}} \quad \ldots\ldots (1)$$

Where W and T are words and tags respectively.

This approach estimates only the probabilities of words with its corresponding tag rather than considering the probability estimations of tag with its previous tags in the given corpus.

The main problem of Maximum Likelihood Estimation (MLE) method is it does not consider local contextual information to assign the most appropriate tag for a given word or sentence. It rather picks the most frequent tag from a given word.

Viterbi Algorithm

Once we have a probabilistic model, the next challenge is to find an effective algorithm for finding the maximum probability tag sequence given an input. It should be clear that an algorithm that enumerates every possible path will run in exponential time, N^T possible path sequences of length T given N possible tags. Fortunately, we do not need to enumerate these paths because the nature of the model allows us to construct a dynamic programming algorithm, which for HMMs is called the Viterbi Algorithm.

This can be explored most intuitively by mapping the problem to an HMM in which the categories "c_i" become the states, the category bigrams become the transition probabilities, and $P(w_i/c_i)$ are the output probabilities. Given all these probability estimates, we can now return to the problem of finding the sequence of categories that has the highest probability of generating an observed sequence of outputs.

This algorithm is an efficient method to find the optimal sequence of states given an observation and used for implementing the tagger. The algorithm optimizes the tagging of a sequence, making the tagging much more efficient in both response time and memory consumptions of the corpus during training and testing of the prepared corpus.The key that makes this algorithm efficient is that we only need to know the best sequences leading to the previous word because of the Markov assumption. According to [9]Viterbi algorithm uses three main steps in order to perform tagging processes which are described as follows:

Given word sequence $Word_1$, . . ., $word_T$, lexical categories Tag_1, . . ., Tag_N, lexical probabilities P(**$Word_1/Tag_t$**), and bigram probabilities P(**Tag_t/Tag_{t-1}**), find the most likely sequence of word class categories C_1, . . ., C_T for the word sequence.

- ✓ The initialization step of Viterbi Algorithm: This step is used to initialize array variable, Score and BackPtr. "Score" temporarily holds the probabilities of words in a given sentence which is going to be tagged and initialized with the product of probabilities of categories at the beginning of sentence (Tag_i/$) and the beginning of word's probability tagged with the given category ($word_1/Tag_i$) in a given sentence. Variable "**BackPtr**" is used to hold the index of the highest probability of the given word.

The syntax of the step is described as follows.

Let T = # of part-of-speech tags W = # of words in the sentence

for t = 1 to T

Score(t, 1) = P($Word_1/Tag_t$) * P(Tag_t/Tag_{t-1})

BackPtr(t, 1) = 0;

- ✓ The iteration step of Viterbi Algorithm: This step is used to determine the lexical probabilities of possible tags for the word for all the rest of words in the sentence. This step is then combined with the contextual probability for each tag to occur in a sequence preceded by the one previous tag. This process continues for all sequences of words in a sentence. The syntax of the step is described as follows.

Let T = # of part-of-speech tags W = # of words in the sentence

for w = 2 to W

for t = 1 to T

Score(t, w) = P($Word_w/Tag_t$) * MAXj=1, T (Score(j, w-1) * P(Tag_t/Tag_j))

BackPtr(t, w) = index of j that gave the max above

- ✓ The sequence step of Viterbi Algorithm: This step is used for tagging processes for each word depending on the information of BackPtr variable. It processes through iterating the

BackPtr that holds the pointer of appropriate category for each word in the sentence. The syntax of this step is described below.

Let T = # of part-of-speech tags W = # of words in the sentence

Seq(W) = t that maximizes Score(t, W)

for w = W-1 to 1

Seq(w) = BackPtr(Seq(w+1), w+1)

HMM (Hidden Markov Model) Method

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved or the hidden states. It is the most widely used method under stochastic approach. It is a statistical Markov model in which the system being modeled is assumed to be moved from state to state (Markov process) with unobserved state. In markov model, the state is directly visible to the observer. In case of HMM, the state is not directly visible to the observer but the output that depends on the hidden state is visible. A discrete Markov model runs through a sequence of states emitting signals. If the state sequence cannot be determined from the sequence of emitted signals, the model is said to be HMM. It is characterized by the following main criteria's [17]:

- ✓ A finite set of states each of which is associated with a probability distribution
- ✓ Transitions among the states are governed by a set of probabilities called transition probabilities.
- ✓ In a particular state an outcome or observation can be generated according to the associated probability distribution. The observation is visible and the states are hidden to the observer and hence the name is Hidden Markov Model (HMM).

HMM is defined formally as a set {S, O, A, B and Π}, where [17]:

- ✓ **S,** which represents the set of N states
- ✓ **O** ,represents the set of observation symbols
- ✓ A, $\{a_{ij}\}$ is a set of state transition probabilities represented in transition probability matrix in which each a_{ij} represents a probability of moving from state Si at time t into state Sj at time t+1.

The state transition probabilities can be defined as $a_{ij} = P(S_{i+1} = j|S_i = i)$ for 1<=i<=n where n is the total number of states, a_{ij}>=0 and$\sum_{i=1}^{n} aij = 1$.

- ✓ $B = b_j(k)$, which is a sequence of emission or observation probability distributions in each of the states' S.

$b_j(k)$ is the observation probability of observation k at the j^{th} state.

The emission/observation probabilities $b_j(k)$ can be computed as $b_j(k) = P(O_{i\ =\ k}\ |S_{i\ =j})$ for 1<=j<=n and 1<=k<=m. $b_j(k)$ is the probability of state j taking the symbol O_i and it should be greater or equal to zero.

- ✓ The initial state distribution $\pi= \pi_i$ which is the probability of the first observation at a given state Si.

Generally an HMM is the set containing {S, O,} where:

S = {S1 S2 S3 S4,…, Sn)

O = {O1 O2 O3 O4,…, Om)

λ= {A, B, π}

The goal of HMM tagger is to select the most likely tags t_1, t_2…, t_n associated with those words.

There are two assumptions about HMM model while we are using it.

- ✓ Every word is not related with all the other words and their tags.
- ✓ Every word's probability depends on the N previous tags only.

Based on the above assumption, HMM taggers select order of tag (sequence) that will maximize the formula:

$$P(\text{word/tag}) * P(\text{tag/previous n tag})$$

Where**word** is the word to which we are going to assign **tag**, the probability of that tag to be for that word in the sentence. In HMM, the entire sentence tag sequence is considered rather than individual word. However, for clarity we look at a single word case example.

A bi-gram HMM tagger is the one that produces its tag result $\mathbf{t_i}$for the unknown word based on the previous tag $\mathbf{t_{i-1}}$given word $\mathbf{w_i}$itself and HMM taggers try to find the tag sequence that maximizes the following formula [9]:

$$P(\text{word/tag}) * P(\text{tag/previous tag})$$

Where P(**word/tag**)is the probability of a word being assigned a particular tag from the list of all possible tags for the word (most frequent tag) and P(**tag/previous n tag**)is the probability that one tag follows another (N-gram)

The optimal sequence of parts of speech tags for a given sequence of words in an input sentence to be tagged can be found using the Viterbi algorithm [18].

When using HMM to perform POS tagging, the aim is to determine the most likely tag sequence that generates the words of sentences. In other words, we calculate the sequence of tags (T) given a sentence (W) that maximizes P(W/T). The Viterbi algorithm can be used to find out the most likely tag sequence. HMM tagger generally chooses a tag sequence for a given sentence rather than for a single word. This approach assumes that we are trying to compute the most probable tag sequence (Ť) of tags T = (t_1, t_2, t_n) for a given sequence of words in the sentence W = (w_1, w_2, w_n):

$$\check{\mathrm{T}} = \underset{\mathrm{t}\epsilon\mathrm{T}}{\operatorname{argmax}} P(T/W)$$

Where, $\operatorname{argmax}_{\mathrm{t}\epsilon\mathrm{T}} P(T/W)$is the set of values of "t" for which P(T/W) attains its maximum value and "argmax" tells us that the function returns the tag sequence that maximizes the probability function value. By Bayes law, P (T/W) can be expressed as

$$\mathrm{P(T/W)} = \frac{P(T)P(W/T)}{P(W)}$$

So we choose the sequence of tags that gives

$$\check{\mathrm{T}} = \underset{\mathrm{t}\epsilon\mathrm{T}}{\operatorname{argmax}} \frac{P(T)P(W/T)}{P(W)}$$

Where, $\underset{\mathrm{t}\epsilon\mathrm{T}}{\operatorname{argmax}} \frac{P(T)P(W/T)}{P(W)}$ is the set of values of "t" for which (P(T)P(W/T)/P(W)) attains its maximum value.

$$\check{\mathrm{T}} = \underset{\mathrm{t}\epsilon\mathrm{T}}{\operatorname{argmax}} \mathrm{P(T)P(W/T)}$$

Where, P(T) is the prior probability and P(W/T) is the likelihood probability.

And from the chain rule of probability, we get the following equation

$$\mathrm{P(T)P(W/T)} = \prod_{i=1}^{n} \frac{p(w_i/w_1t_1 \dots w_{i-1}t_{i-1}t_i)}{p(t_i/w_1t_1 \dots w_{i-1}t_{i-1})} * \quad \text{..... (2)}$$

But for a long sequence of words, calculating probabilities like $\frac{p(w_i/w_1t_1 \dots w_{i-1}t_{i-1}t_i)}{p(t_i/w_1t_1 \dots w_{i-1}t_{i-1})} *$ is not an easy task, there is not an easy way to calculate probability for selecting tag to a word given a long sequence of preceding words. This formula can be simplified by using Markov assumption.

The first assumptions of Markov model is that the probability of a word depends only on its tag, i.e.

$$P(\mathrm{w_i/w_1t_1 \dots w_{i-1}t_{i-1}}) = P(w_i/t_i) \text{..... (3)}$$

Next, we make the assumptions that the tag history can be approximated by the most recent two tags

$$P(\mathrm{t_i/w_1t_1 \dots w_{i-1}t_{i-1}}) = P(t_{i-2}/t_{i-1}) \text{..... (4)}$$

From equations (2), (3) and (4), we get the following generalized formula

$$P(T)P(W/T) = P(t_i)P(t_2/t_i) \prod_{3}^{n} P(t_i/t_{i-2}t_{i-1}) \prod_{i=1}^{n} P(w_i/t_i)$$

After generalizing the equations, the best tag sequence can be chosen in order to maximize the above formula.

Now, as usual we can use maximum likelihood estimation from relative frequency to compute these probabilities. We can use sample corpus to find counts of tag sequences of tags "t_{i-2}, t_{i-1}, t_i"and tags "t_{i-2}, t_{i-1}", where "t_i"is the tag "i", "t_{i-1} and t_{i-2}" are previous two tags, and count of "w_it_i," where "w_i"is the word "i" and "ti" is the tag assigned to word "i". The above generalized equation is used to calculate two kinds of probabilities. The first is tag transition probability, which represents the probability of a tag given the previous tag $P(t_i/t_{i-1})$and which can be described as $P(t_i/t_{i-1}) = \frac{count\ (t_{i-1}t_i)}{count\ (t_{i-1})}$.

This model helps the HMM POS tagger to gather the context of words in the training corpus as lexical model only deals with the probability of the word for the given tag. I.e. relying only on the lexical model may degrade the performance of the tagger and hence it is important to take context of words into consideration [1][19], [20]. Word likelihood probabilities, which represent the probability, given a tag that it will be associated with a given word, $P(w_i/t_i)$ which can be described as $P(w_i/t_i) = \frac{\text{count } (w_it_i)}{\text{count } (t_i)}$, Where W_i and T_i are the i^{th} word in the input sentence and the i^{th} tag in the tagset respectively. Here, the lexical probabilities can be calculated using the relative frequencies. The main goal of the lexical model is to prepare lexicon and the lexical probability of each word for each tag in the training set.

For example, let's see how those two probabilities can be computed.

ይታላ/ADJ ባርዳራ/NN ካሳው/NN ጌርክ/NN ይሚ/NN ኑና/VB ።/PUNC

(My father will go to BahirDar on Wednesday.)

In order to determine the lexical probability of noun "**ካሳው**" occurring in a given corpus as "**NN**", we divide the count of "**ካሳው**" with "**NN**" by the total number of nouns (**NNs**) in the corpus. So, if the count of "**ካሳው**" with "**NN**" is 1 in the corpus and the count on all nouns (NNs) is 611, then the lexical probability will be 0.00163666.

In order to determining transitional probabilities for sequences of words, which boils down to calculating the number of times that the event occurs given the occurrence of another event. By using, $\mathbf{P(t_i/t_{i-1}) = \frac{count(t_{i-1}t_i)}{count(t_{i-1})}}$, we can compute the transitional probabilities.

P(ባርዳራi = NN / ባርዳራi-1 = ይታላ ADJ) ≅ Count(ADJ at position i-1 and NN at i) / Count(ADJ at position i-1).

To describe the bi-gram probabilities of HMM tagger, let's consider the following two Awngi language sentences.

ይታላባርዳራካሳውጌርክይሚኑና።።, which means (My father will go to BahirDar on Wednesday.)

ፋቱኽሳእያታሺውዴስድባኖይሚ።።, which means (To get what you want ask God heartily.)

Here the HMM tagger is expected to assign the correct tag for the word **"ይሚ"** with the assumption that all other surrounding words are correctly tagged as follow:

ይታላ/**ADJ**ባርዳራ/**NN**ካሳው/**NN**ጌርክ/**NN**ይሚ/? ኑና/**VB**።/**PUNC**

ፋቱኽሳ/**VB**እያታ/**PRE**ሼውዴስ/**NC**ድባኖ/**ADV**ይሚ/? ።/**PUNC**

In the first sentence the word "**ይሚ**" used as a **noun** and in the second sentence it is used as **verb**.

By using bi-gram approach tag, the word "**ይሚ**" can be assigned by considering the neighboring words and tags from the given sentences. Look at the sequence in the two sentences above for words surrounding "**ይሚ**":

ጌርክ/**NN**ይሚ/?

ድባኖ/**ADV**ይሚ/?

If we are to choose between **NN** and **VB** for word "**ይሚ**" above, we are expected to select the tag that has higher probabilities:

P (NN/NN) P (ይሚ/NN)…….……. (1)

And

P(VB/ADV)P(ይሚ/VB)…….…….. (2)

Bu using the above two equations we have to select the best tag which maximizes it.

Where **P (VB/ADV)** and **P (NN/ADV)** are tag sequence **P (tag/previous tag)** and **P (ይሚ/VB)**, and **P (ይሚ/NN)** are word probabilities **P (word/tag).** Here, we have to consider the likelihood of **P (ይሚ/VB)** and **P (ይሚ/NN)** terms are not asking "which is the most likely tag for the words accordingly". That is, the likelihood term is not **P (VB/ይሚ)** and **P (NN/ይሚ)** respectively. Instead we are computing **P (ይሚ/VB)** and **P (ይሚ/NN)**. Tag sequence **P(NN/NN)** for example,

tells us how much it is probable to get NN(Noun) if the previous tag is NN(Noun) and **P(VB/ADV)** tells us the probability of obtaining VB(Verb) when the previous tag is ADV(Adverb).

After this, assuming that we have probabilities for the above tag sequence in our corpora as:

P (VB/ADV) =0.26356589 **P (tag/previous tag)**

P (NN/NN) =0.19803601

And assume the lexical probabilities of the words are:

P (**ይማ**/NN) =0.00163666 **P (word/tag)**

P (**ይማ**/VB) =0. 00308642

After calculating of the **P (tag/previous tag)** and **P (word/tag),** the bi-gram probabilities of the HMM will calculate the maximum of **P (tag/previous tag)** and **P (word/tag).** Then we will have the following results:

P (NN/NN)*P (ይማ/NN) = (0.19803601)*(0.00163666) =0.00032465

P (VB/ADV)*P (ይማ/VB) = (0.26356589) *(0. 00308642) =0. 00081348

Based on this calculation, the HMM tagger assigns "**ይማ" asVB** which maximizes the above formula.

Therefore, by following the same procedure the HMM POS tagger assigns the appropriate tag for words which maximizes the above formula.

Generally, the lexical and transitional models are the core models that are used in the HMM POS tagger for this thesis work. The HMM strives to find the optimal sequence of parts of speech tags for a sequence of words in an input sentence by using Viterbi algorithm. The tagger gets the lexical probability and contextual probability from the training corpus.

2.3. Related Works

For POS tagging, many related works have been done for foreign languages like Arabic, Sinhala, Persian and Indian etc. and for local languages like Amharic, Afaan Oromo, and Tigrigna. These languages have several parts of speech taggers that use different mechanisms. This section describes some of those related POST works in local and foreign languages.

2.3.1.Previous Works on Local Languages

According to [1], the researchers have developed parts of speech tagger for Afaan Oromo language by using HMM. In this work, they have used HMM approach for developing the tagger and they have collected 159 sentences (with a total of 1621 words for both training and testing purposes) from different sources to make the corpus balanced and they have used 17 tagsets.

For tagging process, they have used two phases in order to assign word classes to a given Afaan Oromo text. The first phase of the tagger trains on the training data in order to compute and store the lexical and transitional probabilities of training data by using unigram and bigram models of the Viterbi algorithm by taking the stored information and the second phase of the tagger accepts untagged Afaan Oromo texts and tokenized into words. After this, the tagger assigns the correct POS tag for each of the tokenized words. The performance of the tagger has tested using tenfold cross validation mechanism and they got an accuracy of 87.58% and 91.97% for unigram and bigram models respectively. Finally, they have recommended other researchers to develop parts of speech tagger for other local languages by using the same approach.

According to [5], the paper presents POS tagging experiments conducted with the aim of identifying the best method for under-resourced and morphologically rich languages with a corpus that consists in 8,075 tagged sentences or 205,354 tagged tokens by using methodologies like SVM, CRF, TnT and MBT with performances of 94.33%, 94.37%, 93.34% and 92.98% for

known words and 82.26%, 80.66%, 68.96% and 80.51% for unknown words by using 100% of the training data respectively . The result of their POS tagging experiment for Amharic showed that MBT is a good tagging strategy for under-resourced languages as the accuracy of the tagger is less affected as the amount of training data increases compared with other methods, particularly TnT. But the researchers didn't compare the experimental results by using HMM POS tagger. Here, the HMM POS tagger may have comparable performance.

According to [3], the researchers have developed parts of speech tagger for Tigrigna language by applying a hybrid (which is a combination of Brill transformation-error driven learning and HMM) approaches. He has collected a total of 26,000 words from Tigrigna news broadcasting agencies and annotate manually with their corresponding word class and 75% (20,000) of the words were used for training purpose and the remaining 25% (6000) of it used for testing purpose. In addition to this he has identified 36 tagsets for the entire tagging process. This study finds tag of a word from raw text in two main steps. The first step is performed by the HMM tagger and it first annotates the given raw text and provides a level of confidence (threshold value) for each tag sequence. The second step is performed by comparing the confidence level of each tag sequence with the minimum confidence level that is set by the researcher using the output analyzer module. During those steps, if the confidence level is less than that of the minimum confidence level, a window size of two (bigram of the word) is given to the rule based tagger for correction. Otherwise, it is treated as a correct tag. He conducted different experiments for the three types of taggers namely HMM tagger, rule based tagger and hybrid tagger in order to test the performance of the tagger that he has developed. Finally, he has got an accuracy of 89.13% for HMM, 91.8% for rule based and 95.88% for hybrid tagger.

According to [21], the researchers have developed parts of speech tagger for Kafi Noonoo language by applying a hybrid (which is a combination of Brill transformation-error driven learning and HMM) approaches. They have collected a total of 354 untagged sentences from two different genres and annotated using an incremental corpus preparation approach. After assigning word class information on each word within the sentences, both HMM and rule-based taggers are trained on 90% of the tagged sentences to generate probabilities i.e. lexical and transitional probabilities for the statistical component of the hybrid tagger and set of transformation rules for the rule-based component of the hybrid tagger. Both the rule-based and HMM taggers have been

trained on 90% of the tagged sentences. In addition to this, they have identified 34 tagsets for the entire tagging process.Finally, they have got an accuracy of 77.19% for HMM, 61.88% for rule based and 80.47% for hybrid tagger.

According to [13], the researchers have experimented on the use of one of the state of the art probabilistic model for sequence classification, MEMM, to tag Afaan Oromo texts according to the lexical category. This model assigns the correct tag or parts of speech to each word based on the context of the sentence, considering many features. Finally, by using 452 sentences (total word of 6094), they have got an accuracy of 93.01% which was evaluated by tenfold cross validation.

2.3.2.Previous Works on Foreign Languages

According to [22], the researchers have developed parts of speech tagger for Arabic language by using Hidden Markov Model (HMM). They have analyzed the language systematically and come up with tagsets of 55. After this, they have used Buckwalter's stemmer to stem Arabic corpus and they manually corrected any tagging errors. Finally, they have achieved high performance of 97%.

According to [23], Part of speech Tagger for Romanian using Hidden Markov Model. They have used second order (trigram model) Viterbi algorithm to implement the tagger. The corpus used in the experiments and evaluation reported was made of the integral texts in two books: Orwell's 1984 and Plato's The Republic. The amounts of corpus from each source are 117910 and 136960 respectively.

The training and testing processes has been done three times. The first training was done on 90% of “1984“, the second on 90% of “The Republic” and the third on the concatenation of the texts used in the first two (90% of each of the two books). The resulting language models were used to test the corresponding unseen 10% of the texts. As a result, 97.82%, 96.10%and 95.63% performance analysis were reported.

According to [24], Persian parts of speech tagger based on Hidden Markov Model, by using lexicon of 61,521 entries and 64,003 trigrams are used as the language model. They had used the

Festival software for the implementation and make use of Viterbi Decoder provided by Edinburg Speech Tools. The average overall accuracy of the tagger is 95.11%. The accuracy of the known and unknown words is 96.136% and 60.25% respectively.

According to [25], the researchers propose an efficient and accurate POS Tagging technique for Arabic language using hybrid approach that is, a Hidden Markov Model (HMM) integrated with Arabic Rule-Based method. The proposed technique uses the different contextual information of the words with a variety of the features which are helpful to predict the various POS classes. To evaluate its accuracy, the proposed method has been trained and tested with two corpora: the Holy Quran corpus and Kalimat corpus for undiacritized Classical Arabic language. The experiment results demonstrate the efficiency of their methods for Arabic POS tagging. In fact, the obtained accuracy rates are 97.6%, 96.8% and 94.4% for respectively their hybrid tagger, HMM tagger and for the rule-based tagger with Holy Quran corpus. And for Kalimat corpus they obtained 94.60%, 97.40% and 98% for respectively Rule-Based Tagger, HMM Tagger and their Hybrid Tagger.

The following table 1 summarizes related works of local and foreign languages for parts of speech tagger.

No.	**Related Works**	**Objective of the study**	**Used Methodology**	**Total No. of collected Sentences/words**
1	Part of Speech Tagging for Afaan Oromo Language	Investigate the possibility of designing and developing an automatic part of speech tagger for Afaan Oromo language	Uses HMM approach It uses 10-fold cross validation mechanism. Identified 17 tagsets	Collects 159 sentences with word of 1621
2	Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages – The	Identifying the best method for under-resourced and morphologically rich languages	Uses SVM, CRF, TnT and MBT	Collects 8075 sentences with tagged tokens of 205354

	Case of Amharic			
3	Part of Speech Tagger for Tigrigna Language	Develop Part of Speech Tagger model for Tigrigna and analyze the performance of the model	Uses a combination of Brill transformation-error driven learning and HMM approaches and identified 36 tagsets	Collects 26000 words
4	parts of speech tagger for Kafi Noonoo language	Design and develop a POS tagger for Kafi-noonoo language	Uses a combination of Brill transformation-error driven learning and HMM approaches and identified 34 tagsets	Collects 354 sentences from two different genres
5	Automatic Part-of-speech Tagging for Oromo Language Using Maximum Entropy Markov Model (MEMM)	To present probabilistic model for sequence classification, Maximum Entropy Markov Model (MEMM), for tagging Oromo language	Uses MEMM	Collects 452 sentences with total word of 6094
6	parts of speech tagger for Arabic language	To develop parts of speech tagger model for Arabic language	Uses HMM and identified 55 tagsets	-
7	Persian parts of speech tagger based on Hidden Markov Model	To develop parts of speech tagger model for Persian language	Uses HMM	Collects lexicon with 61,521 entries and 64,003 trigrams is used as the language
8	POS Tagging technique for Arabic language	To propose an efficient and accurate POS Tagging technique for Arabic language	Uses Hidden Markov Model (HMM) integrated with Arabic Rule-Based	-

Table 1: summary of related works on POST for local and foreign languages

Chapter Three

Linguistic Properties of Awngi Language

3.1. Overview

Awngi Awi is one of the ten Zones in the Amhara Region of Ethiopia. It is named for the Awi sub-group of the Awngi people, some of whom lives in this Zone. Awngi is a Central Cushitic language spoken by 1.5 million people in an extensive area in northwest Ethiopia, including all of Awi Zone, but also some areas of the Metekel Zone of the Benishangul-Gumuz National Regional State, and various places in the Alefa and K'wara Woredas of the North-Gonder Zone of the Amhara National Regional State [26][27]. It is used as medium of instruction in primary schools and in Enjibara teachers college.

Awngi, similar to Amharic and Tigrigna, uses Geez alphabet. It has twenty-nine consonant phonemes of which five are labialized and six vowel phonemes [28]. Its orthography is syllabic, slightly adopted from the currently used Amharic writing form (**Geez script**) and has four sounds which it does not share with Amharic, namely, ቐ(q),ኈ(ʁ/χ),ኻ(ŋ),and ፀ(ts)[29].

The following tables, table 1 and 2 shows consonants and vowels in Awngi language.

		Labial	Alveolar	Palato-velar		Uvular	
				Plain	Labialized	Plain	Labialized
Plosive	Voiceless	ፕ/p	ተ/t	ከ/k	ኩ/kʷ	ቐ/q	ቕ/qʷ
	Voiced	በ/b	ደ/d	ገ/g	ጉ/gʷ	□/ʁ/χ	ኸ/gʷ
Affricate	Voiceless		ፀ/ts	ቸ/ʧ			
	Voiced		ዘ/dz/z	ጀ/dʒ~ʒ			
Fricative		ፈ/f	ሰ/s	ሸ/ʃ			
Post-stopped fricatives			ኣ/-	ሀ/h			
Nasal		መ/m	ነ/n	ኘ/ŋ	ኜ/ŋʷ		
Flap			ረ/r				
Approximant		ወ/w	ለ/l	የ/j			

Table 2: Awngi language consonants (Source [28][30])

	front	central	back
High	ኢ (i)	እ (ɨ)	ኡ (u)
Mid	ኤ (e)		ኦ (o)
Low		ኣ (a)	

Table 3: Awngi language vowels

In all natural languages, there is a standardized word order in a sentence. For example, in Albanian, Chinese, English, Estonian, Finnish and etc., word orders obey **Subject-Verb-Object** (**SVO**) order. This word order changes to Subject-Object-Verb for German, Dutch, Japanese and Amharic languages [1]. Awngi language word order also obeys the latter order i.e. **Subject-Object-Verb** (**SOV**). If words of the language do not follow this standardized word order, the sentences may convey vague meaning or totally lose their meanings. So clearly understanding of this syntactic structure of sentence can help us to know the relationship between words which in turn leads us to categorize them correctly.

3.2. Inflectional Word classes in Awngi language

There are different amount of words in any language. But not all words in the language have the same task. For example, some words may express action, other words express things and other words join one word to another word. Generally, those words are the building blocks of languages. Understanding of different word classes are important in order to know how words can and should be joined together to make sentences that are both grammatically as well as semantically correct and readable. An understanding of the parts of speech is also important for knowing how to correctly punctuate sentences. When we want to build a sentence, we use those different types of word classes. In order to determine the category of the word, linguists use morphological, syntactical and semantics of words as mentioned in [9].

According to [26]Awngi is an inflective language and has nine general categories or classes of words. Those are nouns, pronouns, verbs, adverbs, adjectives, numerals, prepositions, conjunctions and particles. From those word classes, some of are divided into other sub-classes. For example, pronoun class is categorized as personal pronoun, demonstrative and interrogative pronoun.

3.2.1.Noun class of Awngi

Awngi nouns are generally defined as a person, place, or thing; however, ideas are also nouns. Awngi nouns, like other language nouns, are words used to name or identify a person, place or things. For example:

- ✓ Person: **አቺ** (a man), **አለሙ** (Alemu), **አልማዝ** (Almaz).

  Example, **አለሙዱንዚኹኻ።።** (Alemu eats Potato.).
- ✓ Place: - **ዳንግል** (Dangila), **እንጅብሪ** (Enjibara), **ባርዳር** (BahirDar).

  Example, ይታላ**ባርዳራ**ካሳውጌርክይሚኑና።።(My father will go to **BahirDar** on Wednesday.)
- ✓ Things: **ኽን** (a house), **ወንበር** (chair), **ቢሪ** (Ox).

Awngi has some nouns which reduplicate for plural forms without changing its word class.

ስር (sr) =child	ስራስሪ (srasri)=children
ኪሲ (kisi)=priest	ኪሳኪሲ (kisakisi)=priests
ኹና (ǵuna)=woman	ኹናኹና (ǵunaǵuna)=women

Table 4: Reduplicate noun forms of Awngi

3.2.2.Pronoun class of Awngi

A pronoun in Awngi is used to replace a noun or can be used as noun. In Awngi, pronouns are small words that take the place of a noun. We can use a pronoun instead of a noun. Like nouns, Awngi pronouns inflect for cases except for nominative which is unmarked. The pronouns in Awngi are**:-አን** (I), **እንት** (you), **አኒ** (he, she), **እኖጂ** (we), **እንቶጂ** (you), and**ናጂ** (they). And pronouns can be used in sentences like**:-ናጂዊዱና።።** (**They** finished.)

3.2.3.Verb class of Awngi

Awngi verbs describe action or a state of being. This is the most important part of a speech, for without a verb, a sentence would not exist.

Some examples of verbs are**:-ትንክፍ** (to push), **አንቤብ (**to read), **ይሚ** (to ask).

Example, ሩቱኽሳእያታሼውዴስድባኖይሚ።። (To get what you want **ask** God heartily.)

3.2.4.Adverb class of Awngi

Similar to adjectives, Awngi adverbs can modify a verb, an adjective, or another adverb. It qualifies or modifies a verb, adjective and other adverbs. In Awngi modifiers of verb or verb phrase usually express time, location, manner, etc.

Some examples of adverbs are: **አይቻ** (yesterday), **ልቡናማጂ** (slowly) **እንማቺ** (nearly), **ቻ** (tomorrow).

Example, ግኒልቡናማጂእዘግና።። (he is walking **slowly**.)

3.2.5.Adjective class of Awngi

Awngi adjective modifies (limits or describes) a noun or a pronoun accordingly. Essentially, it provides more information about a person, place, or thing. Awngi adjectives are words that describes, identify or quantify a word by preceding the noun or pronoun which it modifies. Some examples of Awngi adjectives are: **አዊ** (sunny), **ሴኔፍ** (lazy), **ድማ** (red), **ሊጊሲማ** (tall).

Example, **ሴኔፍ**እንዋይይንቱኻ፡፡ (the **lazy** boy came.)

3.2.6.Preposition class of Awngi

Preposition in Awngi shows the relationships between a noun or pronoun and some other words in sentences. This relationship is spatial, temporal, or directional. In Awngi prepositions are words usually coming in front of, a noun or pronoun and express source, destination, location and relation to another word or element. Some examples of Awngi prepositions are: - **ሊ** (with), **ኩክራኽ (**under**), ይኺቺ** (only), **ማንች** (much), **ዴስ** (from).

Example, አቺእማዴስ**ኩክራኽ**፡፡ (everyone is **under** the law.)

In Awngi language prepositions can be attached with nouns and form other word classes. In the following example the preposition **"ዴስ"** means "**from**" is attached with noun **"ቻግኒ"** means "Chagni" and the words form noun with preposition (**NPRE**) word class which means, ቻግኒ**ዴስ**"**from** Chagni" **(ቻግኒ**=noun and **ዴስ**=preposition) .

Example, እንጻይቻግኒ**ዴስ**ይንትኾ፡፡ (he comes **from** chagni)

3.3. Tags and tagsets of Awngi

The broad categories of Awngi word classes are explained in the previous sections by considering the work of [30]. The actual tagsets used in this thesis work will be discussed in this section. Tags are the labels used for adding more information concerning the lexical as well as the transitional category of each word in a sentence and tagsets are the collection of the tags used for developing parts of speech tagger for languages. As far as the researchers' knowledge is concerned, there is no readymade tagsets, unlike that of the Amharic, Affan Oromo, Tigrigna and

Kafi Noonoo languages for Awngi that researchers can make use of it. This implies that identifying and developing tagsets for this thesis work is mentioned to be very important. As a result, the researcher has made continues discussion with Awngi language professionals and teachers at Enjibara teachers college in order to prepare the appropriate tagsets for this language.

The tagsets that are discussed below are classified as a basic class and subclasses of the basic class where Awngi nouns, pronouns, verbs, adjectives, prepositions, conjunctions and adverb are considered to be the basic word classes. In addition, numerals and punctuations are also included as basic word classes in the process of identifying the tagsets.

3.3.1.Nouns and sub-classes of it in Awngi

Awngi nouns have different attributes like numbers, genders and definiteness which can be common noun (like that of **ሆቴል** (Hotel)፣**ቢቲ** (House)), abstract noun (like that of **ዲኪት** (Jacket)), proper noun (like that of **ዝንግኒአይቾ** (Lake Zengena)፣**እንጅባሪ** (Enjibara)፣**ዳንግል** (Dangla)) and concrete nouns (may be the names of place, thing and people, for example, **አለሙ** (Alemu)፣**ወንበር** (Chair)).

Due to tagset complexity problem, we did not include the entire attribute except for nouns. In this main class we identify noun as a general class and other derived classes of noun are:-

noun with conjunction, noun with preposition, noun with auxiliary verb, and noun prepositions and adjectives, noun with prepositions and conjunctions as a sub-classes. This class and its sub-classes are explained in the following examples.

- ✓ Nouns that are attached with conjunctions and which cannot be separated are classified as noun conjunction and are tagged as NC. Words like **ኹናኹኑ-ስታ (**female and**), ክንቲዉ-ስታ (**education and**),** and **ክንቲቫ-ስታ (**mayor and**) etc.** wordscan be tagged as NC.

- ✓ Nouns that are attached with prepositions and which cannot be separated are classified as noun prepositions and are tagged as NPRE. Words like **እምፕልታቾ-ስ (**for social**),ቱሪዝሙዉ-ስታ (**as turisms**),** and **እንፀኽዴ-ስ (**from job**)** etc. words can be tagged as NPRE.

- ✓ Nouns that are attached with auxiliary verbs and which cannot be separated are classified as noun auxiliary verbs and are tagged as NAUXV. Words like **እንግዳ-ኽ (**is Engda**)** etc.and other names which come with auxiliary verbs can be tagged as NAUXV.

- ✓ Nouns that are attached with prepositions and adjectives and which cannot be separated are classified as noun preposition adjectives and are tagged as NPREADJ. Words like **ዋኽዴስሺሾ (ዋኽዴ=noun, ስ=preposition and ሺሾ=adjective)** etc. and other words can be tagged as NPREADJ.

- ✓ Nouns that are attached with prepositions and conjunctions and which cannot be separated are classified as noun preposition adjectives and are tagged as NPREC. Words **like ኬቴም-ዳ-ስታ (**in urban and**), ፄዉሻ-ስ-ስታ (**to control and**), and ድፃሻ-ስ-ስታ (**to destroy and**)** etc. words can be tagged as NPREC.

- ✓ Pronouns that cannot be classified in one of the above classifications are tagged as NN. Words like **ዳንግል** (Dangila), **እንጃብሪ** (Enjibara), **ባርዳር** (BahirDar), **አቺ** (a man), **አለሙ** (Alemu), and**አልማዝ** (Almaz) etc. words can be tagged as NN.

3.3.2.Verbs and sub-classes of it in Awngi

As described in the above section, verbs are possibly the most important part of any text almost in any language in order to make sentences transfer full information. A sentence without a verb cannot give a complete meaning. The subclasses of Awngi verbs are explained as follows.

- ✓ Verbs that are attached with conjunctions and which cannot be separated are classified as verb conjunctions and are tagged as VBC. Words like **ታምባ-ቲታ-ስታ (ታምባ=happen, ቲታ=not, ስታ=and)** which means (not to happen and),**ቻቤልኺ-ስታ (ቻቤልኺ=reciving each other, ስታ=and)** which means (reciving each other and) etc. words can be tagged as VBC.

- ✓ Verbs that are attached with prepositions and which cannot be separated are classified as verb preposition and are tagged as VBPRE. Words like **ኑኑዉስታጊ (**as they said**)** and etc. words can be tagged as VBPRE,

- ✓ Verbs that cannot be classified in one of the above classifications are tagged as VB. Words like **ትንክፍ** (to push), **አንቤብ (**to read), **and ይሚ** (to ask) etc. words can be tagged as VB.

3.3.3.Adjectives and sub-classes of it in Awngi

Awngi adjectives are other word categories that are meant to add extra information to nouns. The class and its sub-classes of adjectives are explained as follows.

- ✓ Adjectives that are attached with conjunctions and which cannot be separated are tagged with ADJC. Words like **ግልፅ-ስታ (**neatness and**), ክራንቲ-ስታ (**mortal and**)** etc. words can be tagged as ADJC.

- ✓ Adjectives that are attached with prepositions and which cannot be separated are tagged with ADJPRE. Words like **አንቤባን-ቲ-ዴስ (**from readers**)** and etc. words can be tagged as ADJPRE.

- ✓ Any other adjective which does not belong to these subcategories is tagged as a general tag ADJ. words like **አዊ** (sunny), **ሴኔፍ** (lazy), **ድሚ** (red), **ሊጊሲሚ** (tall) etc. words can be tagged as ADJ.

3.3.4.Adverbs and sub-classes of it in Awngi

In Awngi language adverbs are words that qualify or modify a verb, adjective or other adverbs. The class and its sub-classes of adverbs are explained as follows.

- ✓ Adverbs that are attached with conjunction and which cannot be separated are tagged as ADVC. Words like **እምቢታማ-ስታ** (**እምቢታማ**=by being fast, **ስታ**=and) which means (by being fast and), **ዴሜካታ-ስ-ታ** (**ዴሜካ**=add, **ታ**=to, **ስታ**=and) which means (to add and) etc. words can be tagged as ADVC.
- ✓ Adverbs that are attached with preposition and which cannot be separated are tagged as ADVPRE. Words like **ካላዉ-ስታ** (**ካላዉ**=he can, **ስታ**=as) which means (as he can), **እንቤብሾ-ዴስ**(from readers) etc. and other words can be tagged as ADVPRE.

- ✓ Other forms of adverbs that cannot be classified under the above classifications are tagged as ADV. Words like **አይኝ** (yesterday), **ልቡናማጂ** (slowly) **እንማቺ** (nearly), **ቻ** (tomorrow) etc. can be tagged as ADV.

3.3.5.Prepositions and its sub-classes of it in Awngi

- ✓ Prepositions by themselves will not give any meaningful information unless they are attached or used with other word classes. In this work we identify preposition tagsets to be tagged as PRE. Words like **ሊ** (with), **ኩክራኽ** (under), **ይኺቺ** (only), **ሚንች** (much) etc. words can be tagged as PRE.

3.3.6.Conjunctions and sub-classes of it in Awngi

Like that of prepositions, conjunctions are words that are either attached or used with some words and serve to connect words, phrase, clauses or sentence. They coordinate words, phrases, clauses and sentences.

Some example conjunctions in Awngi are: - ኾናኾኑ-**ስታ**ቹትካዉ (Female's **and** Mother's).

In the above sentence conjunction "**ስታ**" is used to join two nouns (Female's and Mother's) and used to form another word class "'ኾናኾኑ**ስታ**". In this word "'**ኾናኾኑ**'"'female's"is noun and conjunction "**ስታ**" "and" act as conjunction which comes with noun and forms word class of "Noun with conjunction (**NC**)".

If the conjunctions are used with words such as nouns, adjectives as a separate word they are tagged as CON.

3.3.7.Numerals in Awngi

Awngi numerals are words representing numbers in forms of integers, decimals or can be expressed in words. The numerals can be classified as number, ordinal and cardinal numbers. The Awngi numbers can be re-written in integer or decimal forms. Some examples of Awngi numbers are:-**100, 17.5** etc. The cardinal numbers are numbers like **ላኹ (one), ላኝ (two), ሹኻ (three), ሴዛ (four), እንኳ (five), ዋልታ (six)** and the corresponding ordinal numbers

are**እምፐላንቲ(first) ላቻንቲ (second) ሹካንቲ (third), ሴዛንቲ (fourth), አንኳንቲ (fifth), ዋልታንተ (sixth)** etc. Awngi numbers, cardinal and ordinal numbers can be tagged as NUM, CARDN and ORDN respectively.

3.3.8.Punctuations in Awngi

All Punctuation marks in Awngi language such as: -: ፣, ፡-, „ and “” are tagged by PUNC.

The prepared tagsets that are used in this thesis work are summarized as follows in table 4.

No	Tag	Description	Example
1	ADJ	Represents Adjective tagset.	አቶ/ato
2	ADJC	Represents non-separated Adjective with Conjunction tagset.	ጉቡልትቻኺዉስታ/gubultŋɨŋista
3	ADJPRE	Represents non-separated Adjective with Preposition tagset	አንቤባንቲዴስ/anbebantides
4	ADV	Represents Adverb tagset.	ሼዉቻ/ʃewuŋɨ
5	ADVC	Represents non-separated Adverb with Conjunction tagset.	ዝኮዉስታ/zkowusta
6	ADVPRE	Represents non-separated Adverb with Preposition tagset.	እንቤብኙዴስ/enbebŋudes
7	AUXV	Represents Axillary verb tagset.	አኽኾ/aġɨń
8	CADN	Represents Cardinal number tagset.	ላቻ/laha
9	CON	Represents conjunction tagset.	የኳኒስ/yegʷanis
10	NAUXV	Represents non-separated Noun with Axillary verb tagset.	እንግዳኽ/engdaġɨ
11	NC	Represents non-separated Noun with Conjunction tagset.	ግባቶስታ/gbatosta
12	NN	Represents Noun tagset.	ታደሰ/Tadese
13	NPRE	Represents non-separated Noun with Preposition tagset.	ቴኬምስታኺዳ/tekemstaŋida
14	NPREADJ	Represents non-separated Noun, Preposition and	ዋኽዴስሺሾ/waġɨdesʃɨʃɨo

		Adjective tagset.	
15	PREC	Represents non-separated Preposition with Conjunction tagset.	ዙሪዳስታ/zuridasta
16	NPREC	Represents non-separated Noun, Preposition and Conjunction tagset.	ምክንትዴስስታ/mkntdessta
17	ORDN	Represents Ordinal number tagset.	ላቻንቲ/lahanti
18	PRE	Represents Preposition tagset.	ሊኽዴስ/liġɨdes
19	PUNC	Represents Punctuation mark tagset.	::
20	VB	Represents Verb tagset.	ዴሜካ/demeka
21	VBC	Represents non-separated Verb with Conjunction tagset.	ክንፀየንታካስታ/kn͡tssantakasts
22	VBPRE	Represents non-separated Verb with Preposition tagset.	ዛራዘርኻዴስ/zarazarŋɨdes
23	NUM	Represents Number tagset.	1

Table 5: Awngi Tagsets

Chapter Four

Implementation and Performance Analysis of Awngi POS Tagger

4.1. Introduction

This chapter deals with the detail explanation of designing and implementation of the HMM POS tagger architecture, corpus preparation, experimenting and evaluation of the results as well as performance of the HMM POS tagger for Awngi language. The HMM based POS tagger is customized and adopted for Awngi language. Python is used to test and evaluate the Awngi corpus on HMM POS tagger. The reason behind the choice of Python as testing and evaluating tool is that, it supports many tasks in natural language processing (like parts of speech tagging, morphological analyzer etc.). It is also simple and open source used for different tasks of natural language processing applications (NLPAs).

Assigning grammatical categories to words in a text is an important component of a natural language processing (NLP) system. Text collection tagged with parts of speech (POS) information are often used as a prerequisite for more complex NLP.

Awngi POS tagger is a program that assigns parts of speech to Awngi language words according to the context of that word in a sentence to disambiguate the function of that word in the specific context. We have identified 23 (twenty-three) tagsets for experimental purpose and the implementation of the study is based on those tagsets.

The following figure (1) shows the HMM tagger trainer model and implementation process. A supervised learning method is used for training the HMM model i.e. the training corpus is parts of speech annotated Awngi texts. The tagged corpus is an input to the model. The tagged sentences are given to the tokenizer for tokenizing each sentence to a word level.

After each sentence is tokenized into words, the lexical and contextual models compute the lexical and contextual probabilities which are important for finding a sequence of parts of speech tags for the sequence of words in the input sentence.

After the tagger is trained, it is used for annotating untagged sentences which can in turn be evaluated against the manually tagged data (reference data) of the input testing sentences. Afterwards, the Viterbi selects an optimal parts of speech tag sequence for the given word sequences and gives the tagged word sequences as an output.

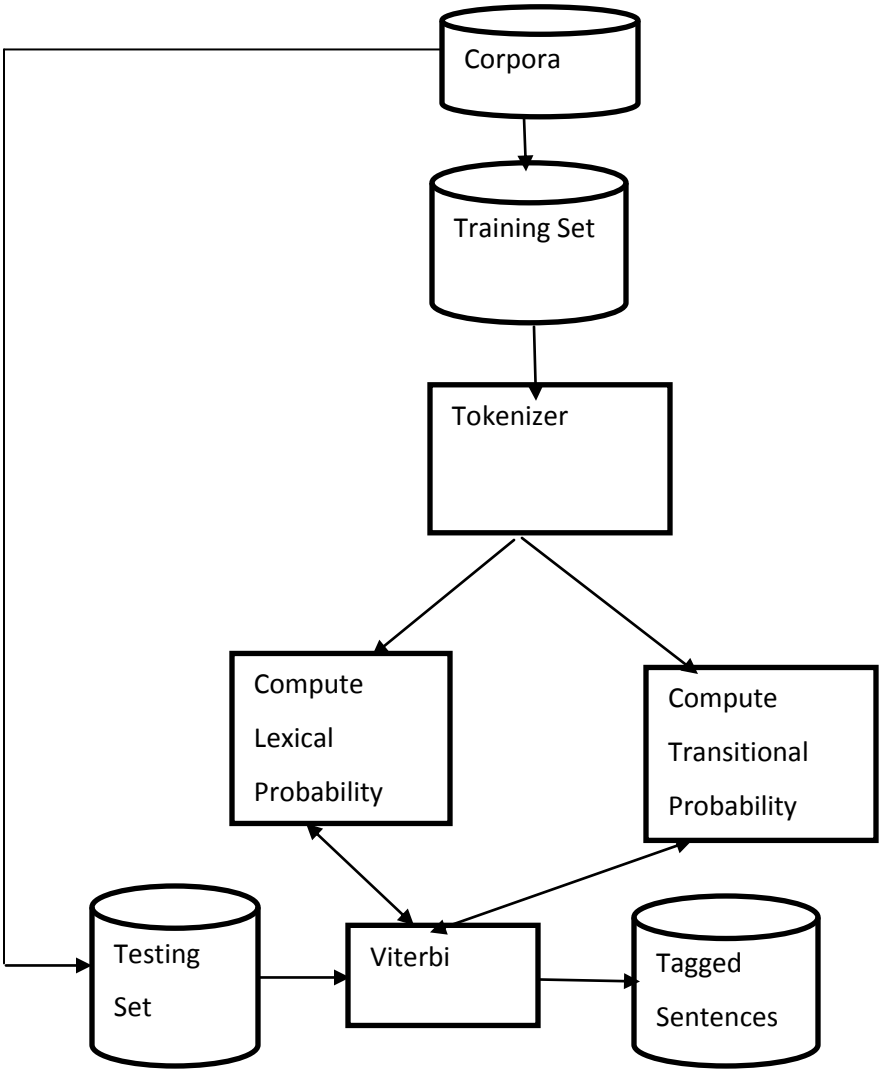

Figure 1: Architecture of the HMM tagger trainer model and implementation processes

4.2. Implementation of the Awngi POS Tagger

4.2.1. Corpus Preparation

Corpus, plural corpora, is a collection of text. It is a collection of texts or speech stored in an electronic machine-readable format [31]. It is also a fundamental tool for any type of research on natural language processing. The corpus with additional linguistic information can be called as annotated or tagged corpus. Such linguistic information in the annotated corpus can be parts of speech information, sentiment information that specifies the word's word class category and sentiment category respectively. The annotated corpus can be used in many NLP applications like parts of speech tagger training and testing, parsing, sentiment analysis etc. In this thesis work, the annotated corpus used is considered to be a text tagged with the corresponding parts of speech tags. In order to process natural language processing tasks like that of parts of speech tagger balanced corpus is required. It is a corpus that represents the words that are used in a language [1]. However, a category specific corpus contains words that are mostly used in that category and if a text from other category to be tagged is given to the tagger trained on this corpus, the performance of the tagger may be degraded. The essence of developing a balanced corpus is, in fact to increase the performance of the tagger when it tags any text taken from any category which implies directly that balanced corpus contains as many words as possible from different categories in their appropriate sense. Larger size of the corpus provides greater learner tendency for the system. The numbers of unknown words are decreased, which results in increasing the accuracy of the system [4].

For this particular study, corpus was collected from different sources such as:

- ✓ From popular Awngi language newspapers which is so called "shirbewa (ሸርቤዋ) newspaper" (which contain social, political, economic and religious aspects) that counts 75% of the whole collected corpora.
- ✓ From Proverbs that counts 5% of the total collected corpora.
- ✓ From Journals that counts 5% of the whole collected corpora and
- ✓ 15% of the remaining corpora were collected from different teaching materials of the language.

For this study, we have collected 350 sentences (with total word of 3760 both for training and testing sets).

4.2.2. Implementation of the Pre-processing Components

In this thesis work, supervised learning approach is used for the Awngi HMM POS tagger. The tagger takes tagged training corpus as an input which needs to be pre-processed by the sentence tokenizer component.

The tokenizer component prepares the words, in fact during the training phase, the word comprises two parts namely the token or word and its corresponding parts of speech, for finding the statistical properties of words and the parts of speech tags.

4.2.3. Experiment and Evaluation of the HMM POS Tagger

Natural language processing (NLP) systems are designed and developed to perform specific tasks as required and expected by users or other systems. A machine translation system is expected to give a correct translation for a given input. An information retrieval system (for example, search engine) is expected to retrieve correctly ranked relevant documents. Similarly, parts of speech tagger is expected to assign a correct tag to a given instance of word.

In general, for a given input, the NLP system is expected to give a correct output. What constitutes correct output and how we can measure it is, however, not an easy task and so is an active area of research in natural language processing. For example, given that two human translators do not translate the same Awngi text into the same English text or into another language.

In order to evaluate the performance of the tagger, we have used tenfold cross validation. This means, that the data has been divided into ten equally parts. Then ten experiments have been performed. The reported result for such an experiment is the average of the results achieved.

4.2.3.1. Lexicon Analysis

This section discuses about the preparation of corpus (plural corpora) in order to carry out the two basic probabilities (namely, lexical and transitional) for the tagger and the lexicon words are stored as wordlists.

For example, table 5 shows some of the words from wordlist dictionary that are used to derive the lexical and transitional probabilities.

	NN	NC	ADJ	ADVC	VB	CON	NPRE	Others	Total
ግትም	24	0	2	0	0	0	0	….	26
ንፋሰስታ	0	1	0	0	0	0	0	….	1
ወይሚ	0	0	1	0	1	0	0	….	2
ይሚ	1	0	0	0	1	0	0	….	2
እምቢታማስታ	0	0	0	1	0	0	0	….	1
እምፐልታቕስ	0	0	0	0	0	0	1	….	1
ስታ	0	0	0	0	0	8	0	….	8
ብሄረሰብ	2	0	0	0	0	0	0	….	2
አግስቱ	0	0	0	0	6	0	0	….	6
ግትሙ	1	0	12	0	0	0	0	….	13
…	…	…	…	…	…	…	…	…	…
Total	611	54	712	10	324	84	23	1,271	3,091

Table 6: sample of lexicons distribution

As the distributions of lexicons of words and tagsets indicate most of the collected Awngi language corpora are tagged as adjectives and/or most of the collected words are adjectives.

From table 5, we can estimate the lexical probabilities by counting the relative frequencies of every word per category from the whole training and annotated corpus. And here, we can derive the statistical information which are used to develop probabilities from the annotated corpus (lexicons) automatically.

As we have discussed in chapter two, the lexicon probabilities of each word ("w_i") occurrences tagged with tag ("t_i") is counted and divided by the counted number of occurrences of the tag ("t_i").

We can calculate it by using: $P(w_i/t_i) = \frac{\text{count } (w_i t_i)}{\text{count } (t_i)}$

For example, let's calculate the lexical probability of the word "**ግትም**" tagged with NN as the above lexicon table indicates.

Count(**ግትም**, NN) = 24, count(NN) = 611

$$P(\textbf{ግትም}/NN) = 24/611 = 0.03927987$$

Table 6 shows sample lexical probabilities' of words in the corpus

Words with the corresponding tag	Lexical Probabilities
P(ግትም/NN)	0.03927987
P(ግትሙ/ADJ)	0.01685393
P(ወይሚ/VB)	0.00308642
P(እያታ/PRE)	0.06666667
P(እኽሻስ/CON)	0.02380952
P(ላና/CADN)	0.066666667
P(ላናንቲ/ORDN)	0.05263158
P(ሴዛ/ORDN)	0.21052632
P(እስቴ/VB)	0.00617284

Table 7: sample lexical probabilities

Here, we can also calculate the transitional probabilities by considering the information of word class categories preceded by other categories developed from training lexicon corpus.

	$	NN	VBC	NC	PRE	PREC	ADV	ADVC	ADVPRE	ADJ	ADJC	ADJPRE	AUXV	NPRE	NPREC	NPREADJ	ORDN	VBPRE	NUM	CADN	CON	VB	NAUXV	PUNC
NN	94	121	6	16	2	1	49	0	0	259	2	0	3	0	0	1	3	1	6	2	13	25	0	7
VBC	1	8	0	0	0	0	2	0	0	1	1	0	0	0	0	0	0	1	0	0	1	1	0	0
NC	3	12	1	1	1	0	7	0	0	22	0	0	0	0	0	0	0	0	0	0	1	1	0	5
PRE	6	4	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0
PREC	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ADV	23	155	4	6	2	0	122	6	1	131	1	0	0	3	2	0	4	3	2	0	18	21	0	12
ADVC	0	4	1	1	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
ADVPRE	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ADJ	89	124	1	18	5	1	128	2	0	203	17	3	3	10	1	0	1	0	1	3	41	29	0	32
ADJC	0	6	0	0	0	0	4	0	0	4	0	0	0	0	0	0	0	0	0	0	2	1	0	7
ADJPRE	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
AUXV	0	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
NPRE	1	6	0	1	0	0	4	0	0	2	0	0	0	2	2	0	0	0	1	0	1	0	0	3
NPREC	0	2	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
NPREADJ	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORDN	0	2	0	0	0	0	3	0	0	2	0	0	0	0	0	0	7	0	0	1	0	1	0	3
VBPRE	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
NUM	0	1	0	0	1	0	4	0	0	1	1	0	0	1	0	0	0	0	0	0	0	2	0	0
CADN	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CON	10	19	1	1	0	0	23	0	0	20	1	0	0	1	0	0	0	0	0	0	1	7	0	0
VB	4	115	0	5	1	0	136	1	0	34	0	0	5	1	0	0	0	0	1	0	4	14	0	3
NAUXV	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PUNC	1	21	2	5	0	0	18	0	1	28	1	0	0	4	0	0	3	0	0	0	0	217	1	0

Table 8: distribution of (ti/ti-1)

Table 7 is used to determine the following:

- ✓ We can determine the total number of each tag in the training corpus by considering the rows of the table. For example, the total number of tag "NN" is 611 (which is the sum of all nouns in the corpus and used to calculate lexical probability).

- ✓ We can determine the total number of previous (t_{i-1}) tags that comes with the corresponding tag by considering the horizontal parts of the table. For example, the count of tag "NN" which comes as a preceding tag with respect to the identified tagsets in the training corpus is 609 (the sum of all the column values with respect to the tag). This is used to know the structure of the language and distribution of tagsets in the collected corpus.

- ✓ Finally, by using the total count of tags, we can calculate different probabilities accordingly.

For this study, bigram is used due to the amount of data (corpus) that we have collected. In transitional probability, bigram considers the information of the category ("t_i") precede the target category ("t"). This indicates: $\mathrm{P}(\mathrm{t_i}/\mathrm{t_{i-1}}) = \frac{count\ (t_{i-1}t_i)}{count\ (t_{i-1})}$

We can compute transitional probabilities by using the above formula. For example, let's compute the transitional probability of:

$$\mathrm{P}(\mathrm{ADJ}/\mathrm{NN}) =?\ count(\mathrm{ADJ}, \mathrm{NN}) = 124$$

$$count(NN) = 611$$

We can take $count(\mathrm{t_{i-1}})$ from lexicon distribution table which is described above)

Therefore, $\mathrm{P}(\mathrm{ADJ}/\mathrm{NN}) = 124/611 = 0.20294599$

Similarly, we can apply the same procedure to compute other transitional probabilities from the corpus as follows:

Bigram categories	Probabilities
P(ADJ/$)	0.37711864
P(ADJ/NN)	0.20294599
P(VBC/CON)	0.01190476
P(NPREC/ADV)	0.00193798
P(ADJ/VBC)	0.0625
P(NC/CON)	0.01190476
P(NN/$)	0.39406779
P(NUM/PRE)	0.06666667
P(NPRE/$)	0.00423729
P(ADV/VBC)	0.25
P(VB/PRE)	0.06666666
P(PUNC/VB)	0.66975309

Table 9: sample transitional probabilities

As it has indicated in table 8, the code "tag/$" indicates occurrence of the words at the beginning of the sentence. It has been observed that mostly nouns come at the beginning of Awngi sentences than that of adjectives including other tagsets of the language and verb words mostly occurred at the ending of sentence. This shows that, in all training sample corpus noun words register the highest probability of occurrence at the beginning of sentences and Awngi language verbs register the highest probability of occurrence at the ending of the sentences.

This result indicates the structure of sentences in the language is Subject-Object-Verb (SOV).

4.3. Performance Analysis of the HMM Tagger

Many experiments with different training sets have conducted by using the collected corpora for Awngi language HMM POS tagger. In order to perform this, the collected corpora are divided into two main sets namely, training and testing sets. After this, In order to conduct the experiments, we have used tenfold cross validation. Here, to calculate the performance of the

individual test sets, we have considered the count of words in the test set and the correctly tagged words by referencing the gold corpus. The tagger is evaluated by comparing the tagged output with the Gold standard test set.

$$performance = \frac{Correctly\ Tagged\ words}{Count\ of\ words} * 100$$

By using this formula, we can train and test the performances of the tagger for n-grams. In this thesis we have implemented HMM POS tagger only for unigram and bigram taggers due to sparseness of the collected corpus. Here, to evaluate the performance of the tagger by using unigram (n-gram, n = 1) tagger is a simple statistical tagging algorithm. For each token, it assigns the tag that is most likely for that token's text.Before a unigram tagger can be used to tag data, it must be trained on a training corpus. It uses the corpus to determine which tags are most common for each word.

For example, let's compare the following sentences with the gold one.

Gold standard test set1: <s><s>**ADJ ግትሙ** NN ታሲ ADJ ምታኒሊ**ADJ አይሎ** VB ታምትኚ PUNC **::**</s></s>

Gold standard test set2: <s><s> ADV እን ADV አኽሻስኪ**ADJ ግትም** ADV አንቤብኑስ CON አኹኪ ADV ቹኒትኑሳ ADJ እንክራንትካ ADV አንቤባኒስ PUNC ፣ ADV እንክራኒስ PUNC ፣ ADV ቐሊትኒስ PUNC ፣ ADJ ቴግባርካዋ ADV ፌፄማኒስ ADJ እኑ ADV ፄጌራስ VB ካንትኔ PUNC **::**</s></>

As we have seen in the first sentence, the word "**ግትሙ**" which is an adjective ("**ADJ**") and its frequency in the training set is 12 and 1 times as adjective ("**ADJ**") and noun ("**NN**") respectively. Similarly, the word "**አይሎ**" occurs in the training set as adjective and conjunction with frequency of 11 and 4 respectively. Unigram tagger tags the word "**ግትሙ**" as "**ADJ**" and "**አይሎ**" as "**ADJ**" too since both of the words have the highest frequency in the training set and the experimental result is: **<s><s>ግትሙ /ADJ**ታሲ /NN ምታኒሊ /ADJ **አይሎ /ADJ**ታምትኚ /VB **::** /PUNC </s> /</s>

But in the second sentence, the word "**ግትም**" which is an adjective and noun with frequency of 2 and 24 respectively. Here, the tagger tags the word as "**NN**" incorrectly, but must be tagged as

"**ADJ**". This incorrect tag assigning problem comes from that unigram tagger simply picks the most frequent tag without considering contextual meanings of the whole words in the given sentence. The experimental result is: <s><s>እን /ADJ አኽሻስኪ /ADV **ግትም /NN**አንቤብኑስ /ADV አኹኪ /CON ቹኒትኑሳ /ADV እንክራንትካ /ADJ አንቤባኒስ /ADV ፣ /PUNC እንክራኒስ /ADV ፣ /PUNC ቾሊትኒስ /ADV ፣ /PUNC ቴግባርካዋ /ADJ ፌፄማኒስ /ADV እኑ /ADJ ፄጌራስ /ADV ካንትኔ /VB ። /PUNC </s></s>

Generally, unigram tagger assigns a tag to a word which is most likely to occur. More specifically, it trains with a training data and calculates the occurrences of the words, then tags the test data according to the occurrence statistics in the training data.

Let's consider the first test (10% of the training set) set with gold tagged sentence and the unigram tagger performance results that we got from the experiment.

$$\boldsymbol{performance\ of\ Fold1} = \frac{\boldsymbol{Correctly\ Tagged\ words\ of\ Fold1}}{\boldsymbol{Count\ of\ words\ in\ Fold1}} * \boldsymbol{100}$$

$$performance\ of\ Fold1 = 170 * \frac{100}{192} = 88.54$$

By following similar method, we can calculate the performance of the whole tested folds.

Then the accuracy of the tagger is the average of the whole test sets.

$$\boldsymbol{Accuracy} = \sum \frac{\boldsymbol{performance\ of\ tested\ folds}(\boldsymbol{Fold1} + \boldsymbol{Fold2} + \cdots + \boldsymbol{Foldn})}{\boldsymbol{total\ No.of\ tested\ folds}(\boldsymbol{n})}$$

Where, $Fold_1$+$Fold_2$+……+$Fold_n$ is the tested folds and n is total number of performed tests.

Tested on	Count of words	Correctly tagged words	Incorrectly tagged words	Accuracy in percent
Fold 1	192	170	22	88.54
Fold 2	237	198	39	83.54
Fold 3	229	207	22	90.39
Fold 4	164	140	24	85.36
Fold 5	208	182	26	87.50
Fold 6	248	214	34	86.29
Fold 7	239	206	33	86.19
Fold 8	224	179	45	79.91
Fold 9	188	154	34	81.91
Fold 10	150	123	27	82
Accuracy				85.16

Table 10: unigram accuracy of the tagger

Since unigram taggers select the most frequent tag in the training data without contextual consideration, less frequent words in the training set may tag incorrectly. Bigram tagger tags a word with one specific difference of unigram tagger. That means, the tagger checks one word before.

Let's see the following example how bigram tagger works.

Gold standard test set is: **<s><s>**NN ግትም**NC እኑስ ADV አይሎ NN**ኽታዋ ADV ፌቴርስትኹ ADJ አርቱ NN ዜር CON ያኽታ**ADV ዜርፎ** ADJ ፂኒሱንኩ ADJ ማራማርጃንትካ VB እርዳትፃና PUNC **::</s></s>**

Experimental result is: <s><s>ግትም/NN **እኑስ/CON አይሎ/ADJ**ኽታዋ /NN ፌቴርስትኹ /ADV አርቱ /ADJ ዜር /NN ያኽታ /CON **ዜርፎ/ADJ**ፂኒሱንኩ /ADJ ማራማርጃንትካ /ADJ እርዳትፃና /VB **::** /PUNC </s> /</s>

As we have seen in the above sentence, there are words that are tagged incorrectly.

For example, the word "**እኑስ**" is tagged as "**CON**", the word "**አይሎ**" is tagged as "**ADJ**", and word "**ዜርፎ"** is tagged as "**ADJ**" but all words must be tagged as "**NC**", "**ADV"** and "**ADV"** respectively. Then after identifying of correctly and incorrectly tagged words, we can calculate

the performance of Fold1 as follows. To measure the performance of each tested folds and accuracy, we can follow the same procedure as unigram taggers.

$$\boldsymbol{performance\ of\ Fold1} = \frac{\boldsymbol{Correctly\ Tagged\ words\ of\ Fold1}}{\boldsymbol{Count\ of\ words\ in\ Fold1}} * \mathbf{100}$$

$$performance\ of\ Fold1 = 173 * \frac{100}{192} = 90.10$$

This performance calculation shows, the context wise information of words (the appearance of words in the sentence) affects the determination of word categories for the language.

Tested on	Count of words	Correctly tagged words	Incorrectly tagged words	Accuracy in percent
Fold1	192	173	19	90.10
Fold 2	237	211	26	89.03
Fold 3	229	207	22	90.39
Fold 4	164	144	20	87.80
Fold 5	208	185	23	88.94
Fold 6	248	218	30	87.90
Fold 7	239	215	24	89.96
Fold 8	224	191	33	85.27
Fold 9	188	161	27	85.63
Fold 10	150	125	25	83.33
Accuracy				87.84

Table 11: bigram accuracy of the tagger

Finally, the results of the experiments for both unigram and bigram taggers perform with accuracy of 85.16% and 87.84% correctly tagged words in average respectively.

Depending on this accuracy, we can conclude that bigram tagger tags a word by improving the performances of the unigram tagger that it checks for the occurrences of words together with one word before.

Generally, in HMM POS tagger the accuracy depends on the number of N-grams since the probability of getting incorrectly tagged words will be minimized.

Chapter Five

Conclusions and Recommendations

5.1. Conclusions

Natural language is the medium for communication which is incorporated by every human being. One of the most important activities in processing natural languages is parts of speech tagging.

Parts of speech tagger (POS) also called, as grammatical tagging or word category disambiguation, is the task of assigning to each word of a text the proper POS tag in its context of appearance in sentences. It is an initial stage of linguistics, text analysis like information retrieval, machine translator, text to speech synthesis, information extraction etc.

The importance of the problem focuses from the fact that the POS is one of the first stages in the process performed by various natural language related processes. There are different approaches to the problem of assigning a parts of speech (POS) tag to each word of a natural language sentence. Here, we have prepared balanced corpus and a tagset for the collected corpus with the help of linguistic experts. The thesis is about performing parts of speech tagging for the Awngi language using Hidden Markov Model (HMM). We have experimented the collected corpus for both unigram and bigram models with accuracy of **85.16%** and **87.84** respectively.

Bigram tagger tags a word by improving the performances of the unigram tagger that it checks for the occurrences of words together with one word before. The accuracy of HMM POS tagger depends on the number of N-grams since the probability of getting incorrectly tagged words will be minimized.

Due to the limitations of the tagger presented in this thesis work, the results presented herein can function as a first benchmark for future research on parts of speech tagging of Awngi language. Further, tagging data with unknown words is also an essential need to handle in the tagger. When the system reach an unknown word, current tagger fails to propose a tag, since the system is not trained for that word and the tagging algorithm doesn’t have enough intelligence to propose tags for untrained words and tags it as UNKN.

5.2. Recommendations

As it has been discussed in the above chapters, development of HMM POS tagger for Awngi language is in its initial stage. Hence, there are several areas of research for Ethiopian languages and Awngi language in particular that should be recommended for future researchers in the area of natural language processing.

Finally, this thesis work suggests the following items as future work:

- ✓ Standardized and readily available corpus is very important for natural language processing applications. Clearly stating, it affects the accuracy of the work in this area. Therefore, preparation of standardized corpus is recommended for further researches in the area of natural language processing for Awngi language.
- ✓ In this work, a small size of corpus (total of 3760 words and 350 sentences) was used for training and testing the tagger. Therefore, another research that uses large size corpus is recommended to improve the accuracy of the tagger by minimizing unknown and incorrectly tagged words.
- ✓ Another approaches for POS tagging of Awngi language can be used such as neural network, hybrid approach that uses Hidden Markov Model and rule based approach or transformational error driven learning and neural network approach in order to handle the occurrences of unknown words
- ✓ Extending this work so that the tagger identifies word features like genders, numbers, persons, tenses aspects etc. can also be future research areas.

References

[1] Getachew Mamo and Million Meshesha. “Part of Speech Tagging for Afaan Oromo Language.” In Proceeding International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence, Vol.1, No. 3, USA, pp.1-5, 2011.

[2] Grover Hudson. “Linguistic Analysis of the 1994 Ethiopian Census”, Vol. 6, No. 3, pp. 89–107, 2015.

[3] Teklay Gebregzabiher. “Part of Speech Tagger for Tigrigna Language.” Master's Thesis, Addis Ababa University, Addis Ababa, unpublished, 2010.

[4] Mohammed Hussen. “Part of Speech Tagger for Afaan Oromo Language Using Transformational Error Driven Learning (TEL) Approach.” Master's Thesis, Addis Ababa University, Addis Ababa, unpublished, 2010.

[5] Martha Yifiru, Solomon Teferra and Laurent Besacier. “Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages – The Case of Amharic”, pp. 2–5, 2011.

[6] Abraham Gizaw. “Improving Brill’s Tagger Lexical and Transformation Rule for Afaan Oromo Language”, unpublished, 2013.

[7] Sisay Fissaha. “Part-of-speech tagging for Amharic using Conditional Random Fields”. Informatics Institute, University of Amsterdam Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, 2005

[8] Bilel Ben And Fethi Jarray. “Genetic Approach for A Rabic Part of Speech”, Vol. 2, No. 3, pp. 1–12, 2013.

[9] Daniel Jurafsky and James Henery. "Speech and Language Processing", an Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice-Hall, Inc., 2000.

[10] Beatrice Santorini. “Part of Speech Tagging Guidelines for the Penn Treebank Project”, unpublished, 1991.

[11] Lei La, Qiao Guo and Qimin Cao. “Training MEMM With PSO : A Tool for Part of

Speech Tagging”, Vol. 7, No. 11, pp. 2511–2517, 2012.

[12] Nikos Karampatziakis. “Maximum Entropy Markov Models”, unpublished, 2006.

[13] Abraham Tesso, Degen Huang and Xiaoxia Liu. “Automatic Part of Speech Tagging for Oromo Language Using Maximum Entropy Markov Model (MEMM)”, Vol. 10, pp. 3319–3334, 2014.

[14] Anisha Aziz and Sunitha Cheryan. “A Hybrid Parts of Speech Tagger for Malayalam Language”, pp. 1502–1507, 2015.

[15] Beáta Megyesi. “Brill ’ S Rule-Based Pos Tagger”, pp. 1–10, 1992.

[16] Kanak Mohnot, Neha Bansal, Shashi Pal and Ajai Kumar. “Hybrid Approach for Part of Speech Tagger for Hindi Language”, Vol. 4, No. 1, pp. 25–30, 2014.

[17] Gordana Ili´C. “HMM Tagger for Swedish”, pp. 1–6, 2009.

[18] Chrstopher Manning and Hinrich Zchutze. “Foundations of Statistical Natural Language Processing”, MIT press cambridge, 2000.

[19] Johan Hall. “A Probabilistic Part of Speech Tagger With Suffix Probabilities A Probabilistic Part of Speech Tagger With Suffix Probabilities”, unpublished, April, 2003.

[20] Yahya Othman and Mohamed Elhadj. “Statistical Part of Speech Tagger for Traditional Arabic Texts”, Vol. 5, No. 11, pp. 794–800, 2009.

[21] Zelalem Mekuria. “School of Graduate Studies College of Natural Sciences Department of Computer Science Design and Development of Part-of-speech Tagger for Kafi-noonoo Language Addis Ababa University School of Graduate Studies College of Natural Sciences Department of Computer Science Design and Development of Part-of-speech Tagger for Kafi-noonoo Language,” unpublished, 2013.

[22] Fatma Al-Shamsi and Ahmed Guessoum. “A Hidden Markov Model – Based POS Tagger for Arabic”,unpublished, 2006.

[23] Anijat Jayaweera and Nafial Dias. “Hidden Markov Model Based Part of Speech Tagger for Sinhala Language”, Vol. 3, No. 3, pp. 9–23, 2014.

[24] Ali Azimizadeh and Mohammad Mehdi. “Persian Part of Speech Tagger Based on Hidden

Markov Model", pp. 121–128, 2008.

[25] Meryeme Hadni, Said Alaoui, Abdelmonaime Lachkar and Mohammed Meknassi. "Hybrid Part-Of-Speech Tagger for Non-Vocalized Arabic Text," Int. J. Nat. Lang. Comput., vol. 2, no. 6, pp. 1–15, 2013.

[26] Andreas Joswig. "The Phonology of Awngi", SIL International, 2010.

[27] Paul Fallon. "The Velar Ejective in Proto-Awngi," no. 2006, pp. 10–22, 2009.

[28] Andreas Joswig and Hussein Mohammed. "A Sociolinguistic Survey Report ; Revisiting The Southern Awngi Language Areas of Ethiopia", SIL International, 2011.

[29] Debre Markos. "Proceedings of the First National Research Symposium on Sustainable Development : A Great Concern in Africa", 2010.

[30] Tsegaye Misikir."Developing a Stemming Algorithm for Awngi Text: A longest Match Approach", unpublished, 2013.

[31] Pierre Nugues."An Introduction to Language Processing with Perl and Prolog." Springer-Verlag Berlin Heidelberg, Germany, 2006

Appendix

Sample untagged Awngi texts

ዩኒቨተርስቲዉፐሬዘዳንታዶክተርድሳልኝመንግሻእንወክታዳማይኬልአንፅኬጀሜሩሳንደርኬቴምዳስታዙራመሪዳአግስታንኩእምፐልታቑኩቤንካስአገልግሎቴይግስቑትካዉሳስታስራስሪዉሳክርኞኔኬታቫስካለፃዉአኻኞዙሜካ፡፡

ግንባቲካሚንፄስታሲሊትንፄቤንካዋአሚኹአኽኞዙሙንኩዶክተርደሳልኝይጉዋንቲእከምኒዉአክኺዋኻትኟኒዳኒስአገልግሎቴጀሜሬንካ፡፡

ቤረሰቡቑፅቫፂዳጉማግሲሺኩዳድወረዳዉሸፋዝኩዋንቲአቶገዛኽኝስሜነህእየኒዉካንቲስአግፅኑዉሳሼዉቴቴኬምስታቫስድማጋቲታቱትኼካለካልቫስስራስሪክንቲዉማንድዴስይጉዋቲንታታንቦፒፊትኟንኩአኽኞእርዳታፅካ፡፡

ስፔንአጌራዳአግስታዉአሚእስታማእቑስታዉምንግስታዋየኻያሱእርዳትንቲአችትአንኳሚሊዮንብሩእርዳቲስጎነዴርኬቴምዳግቢስታኹካሚንፂማይኬልጎቤንስታ፡፡

ላሊበላዋጎቤናንታካዉቼፍዴሜካማጊያኽፌዩንኩዋልታአርፍካዳአጌራኻስኩስታበርኽቼዉጎቤናንታካዴስ 15
ሚሊዮንብርዴስጀለሚፄኩንፅኞላሊበላኬቴምችፅቫፂዉባይሉ፣ቱሪዝሙዉስታስፖርታፀቫናአቫፅኻ፡፡

ጋያኪሜስኪሉሳባሉካንታኹስቴርትኩፁፍአዋዋዉሳታሪኮባሉሳኪቤርፅችሳቴርቶክችክችስካንፀፃዌኽ፡፡

አንቤኪላዙራሙሪደስታቫንቫናዳፄዉስታንኩቴርትካዋኩልቻማከንፅፃዉቴርትኪላብርኹስትዛኽዝሜድዉካሴትኺዉሊሊቱባል አኽኞጌሌፃ፡፡

ክቻስቱኒናዉደብኪ 15
ጌርካቺፋአቲማፄኑጋዚቲዳኪየኻሶኬትዴስላኺኒፊዄያፊዄቫፄኑሬድዮኑቴርትዳአዋኽዳሚንችካዳሴስታቲንኩኬቤልካዝኩና፡፡

እኒኟዋኪላቲንቲካማጊዝኩኽሳዴሶካፂካማፊዄያፊዄይቫፅኒእምፐልወረዳአለውዴስሚንችዴሶከፃዉአኽኞዉላዙሩሙሊዉሳታምብስታንቶዚኒኬንኪ፡፡

እጃዉጌቤልትኺሆስፒታልዳዋይሚስታያኹኻአለሚቱየኔታሆስፒታልካዉእጁጉቡልትቫኺዉስታፋርማሲዉአገልግሎታእይኺግልፅስታካሲስታንትዝኩኻያኻታቤቴርስትኹእንፃክስትኺእጁማበጀትቱስቲስትክምዳእሼታካላፅቫጌሌፅስትኾ፡፡

ክልሉቲንማንዳኟዉቢሩትክታሊአላፊአቶአሊገበየሁዙሙኑዉስታጊቤዴርኑስአኻኹእንፃኽስታኺእጀበኬንፃዉስታቑንደስቴፃአንጌሌታስታንታፄዋሸኹየኹስእስካዊእንፂኻስታኺዉቴርታዉለጊሁስፒታልካዳንባርዳስርትኹግልፅ፣አዳታፄዉስታኺካላዉስታካሊስታንቶያጋዉእንፃክስትኺዉቴርትቤቴርስታቫእጁዉሳባቴልኞይጉፅቫስዲግስኺዴሜካታስታእጁዉበጀታቱስቲስትክምዳእሼታካሊፃንካ፡፡

Sample tagged Awngi texts

<s><s>ዩኒቨተርስቲዉ /ADJ ፕሬዘዳንታ /ADJ ዶክተር /ADJ ድሳልኝ /NN መንግሻ /NN እንወክታዳ /ADV ማይኬል /NN አንፅኬ /ADV ጀሜሩ /ADV ጎንደር /NN ኬቴምዳስታ /NPREC ዙራመሪዳ /ADV አግስታንኩ /ADJ እምፕልታቹኩ /ADJ ቤንካስ /ADJ አገልግሎቴ /NN ይግስ /ADV ቹትካዉሳስታ /NC ስራስሪዉሳ /ADJ ክርኼ /NN ኔኬታሻስ /ADV ካለፃዉ /ADV አኻኼ /ADV ዙሜካ /VB ፡፡ /PUNC </s></s>

<s><s>ግንባቲ /NN ካሚንፄስታ /ADJC ሲሊትንፄ /ADJ ቤንካዋ /NN አሚኹ /ADV አኽኼ /VB ዙሙንኩ /ADJ ዶክተር /ADJ ደሳልኝ /NN ይጉዋንቲ /ADJ እከምኒዉ /ADJ አክኺ /NN ዋኻትጀኒ /ADV ዳጊስ /ADV አገልግሎቴ /NN ጀሜሬ /ADV ንካ /VB ፡፡ /PUNC </s></s>

<s><s>ቤረስቡ /ADJ ቺፅሻፂዳ /ADV ጉማግሲ /NN ሺኩዳድ /NN ወረዳዉ /ADJ ሸፋ /NN ዝኩዋንቲ /ADJ

አቶ /ADJ ገዛኽኝ /NN ስሜነህ /NN እየኒዉ /NN ካንቲስ /ADV አግፅኑዉሳ /ADJ ሼዉቴ /NN ቴኪምስታሻስ /ADV ድማ /ADJ ጋቲታ /ADJ ቱትኼ /NN ካለካልሻስ /ADV ስራስሪ /ADV ክንቲዉ /ADJ ማንድዬስ /NN ይጉዋቲንታ /ADV ታንቦፒ /ADV ፊትጀንኩ /ADV አኽኼ /ADV እርዳታፅካ /VB ፡፡ /PUNC </

<s><s>ስፔን /NN አጌራዳ /ADV አግስታዉ /ADV አሚ /NN እስታማ /ADV እቹስታዉሸፋ /VB ምንግስታዋ /ADV የኻያሱ /ADV እርዳትንቲ /ADJ አችት /ADJ አንኳ /ORDN ሚሊዮን /ORDN ብሩ /NN እርዳቲስ /ADJ ጎነዬር /NN ኬቴምዳ /ADV ግቢስታኹ /ADJ ካሚንፂ /ADJ ማይኬል /NN ጎቤንስታ /VB ፡፡ /PUNC </s></s>

<s><s>ላሊበላዋ /NN ጎቤናንታካዉ /ADJ ቹፍ /NN ዴሜካማፂያኽ /VB ፊዩንኩ /ADJ ዋልታ /NN አርፍካዳ /ADV አጌራኻስኩስታ /VBC በርኽቹዉ /ADJ ጎቤናንታካዬስ /NPRE 15 /NUM ሚሊዮን /ORDN ብርዬስ /NPRE ጀለ /ADV ሚፄ /NN ኩንፅኼ /ADV ላሊበላ /NN ኬቴም /ADJ ችፅሻፂዉ /ADJ ባይሉ /ADJ ፣ /PUNC ቱሪዝሙዉስታ /NPRE ስፖርታ /ADJ ፀሻና /NN አሻፅኻ /VB ፡፡ /PUNC </s></s>

<s><s>ጋያኪ /ADV ሜስኬሉሳ /ADJ ባሉ /NN ካንታኹስ /ADV ቴርትኩ /ADJ ፁፍ /ADJ አዋዋዉሳ /ADJ ታሪኮ /NN ባሉሳ /ADV ኬቤርፅኙሳ /ADJ ቴርቶ /NN ክችክችስ /ADV ካንፀፃዌኽ /VB ፡፡ /PUNC </s></s>

<s><s>አንቤኪላ /VBC ዙራሙሪደስታ /ADVC ሻንሻናዳ /ADV ፄዉስታንኩ /ADJ ቴርትካዋ /NN ኩልቾማ /ADV ከንፅፃዉ /ADJ ቴርት /NN ኪላ /CON ብርኹስት /ADJC ዛኽዝሜድዉ /ADJ ካሴትኺዉ /ADJ ሊሊቱ /ADJ ባል /NN አኽኼ /ADV ጌሌፃ /VB ፡፡ /PUNC </s></s>

<s><s>ክቾስቱኒ /VB ናዉ /AUXV ደብኪ /NN 15 /NUM ጌርካ /NN ቺፋ /ADV አቲማፄኑ /VB ጋዚቲዳኪ /ADV የኻ /CON ሶኬትዬስ /ADVC ላኺኒ /ADV ፌያፌሻፄኑ /VB ሬድዮኑ /ADJ ቴርትዳ /ADV አዋኽዳ /ADV ሚንችካ /ADJ ዳሴስታቲንኩ /ADJ ኬቤልካ /NN ዝኩና /VB ፡፡ /PUNC </s></s>

<s><s>እኒኝዋኪላ /NN ቲንቲካማጊ /VB ዝኩኽሳ /ADV ዴሶ /NN ካፂካማ /ADV ፌያፈይኻፅኒ /VB እምፐል /ADJ ወረዳ /ADJ አለውዴስ /ADV ሚንቾ /ADJ ዴሶ /NN ከፃዉ /VB አኽኼ /ADV ዉላ /ADJ ዙሩሙሊዉሳ /NN ታምብስታንቶ /ADV ዚኒኬንኪ /VB ፡፡ /PUNC </s></s>

<s><s>እጃዉ /ADJ ጌቤልትኺ /NN ሆስፒታልዳ /ADV ዋይሚስታ /ADJ ያኹኻ /VB አለሚቱ /NN የኔታ /NN ሆስፒታልካዉ /ADJ እጁ /NN ጉቡልትኻኺዉስታ /ADJC ፋርማሲዉ /ADJ አገልግሎታ /NN እይኺ /ADV ግልፅስታ /ADJC ካሲስታንት /ADJ ዝኩኻ /ADJ ያኻታ /NN ቤቴርስትኹ /ADJ እንፃኽስትኺ /ADJ እጁማ /ADJ በጀት /NN ቱስቲስ /ADV ትክምዳ /ADV እሼታ /ADV ካላፅኻ /VB ጌሌፅስትኾ /VB ፡፡ /PUNC </s></s>
<s><s>ክልሉ /ADJ ቲን /NN ማንዳጃዉ /ADJ ቢሩ /NN ትክታሊ /ADJ አላፊ /ADJ አቶ /ADJ አሊ /NN ገበየሁ /NN ዙሙኑዉስታኒ /VB ቤዬርኑስ /ADV አሻኹ /ADJ እንፃኽስታኺ /ADJ እጀ /NN በኬንፃዉስታ /ADJC ቑንደስቴፃ /ADV አንጌሌታስታንታ /ADV ፄዋሸኹ /VB የኹስ /CON እስካዊ /ADJ እንፂካስታኺዉ /ADJ ቴርታ /NN ዉለጊ /ADJ ሁስፒታልካዳ /ADV ንባርዳ /ADV ስርትኹ /ADV ግልፅ /ADJ ፣ /PUNC አዳታ /NN ፄዉስታኺ /ADV ካላዉስታ /ADVPRE ካሊስታንቶ /ADV ያጋዉ /VB እንፃኽስትኺዉ /ADJ ቴርት /NN ቤቴርስታኻ /VB እጁዉሳ /ADV ባቴልኼ /ADJ እጀ /NN በኬንፃዉስታ /ADJC ቑንደስቴፃ /ADV ይጉፅኻስ /ADV ዲግስኺ /ADV ዴሜካታስታ /ADVC እጁዉ /ADJ በጀታ /NN ቱስቲስ /ADV ትክምዳ /ADV እሼታ /ADV ካሊፃ /VB ንካ /VB ፡፡ /PUNC </s></s>

Awngi alphabets and pronunciation

	ኣ(a)	ኡ(u)	ኢ(i)	ኤ(e)	እ	ኦ(o)			ኣ(a)	ኡ(u)	ኢ(i)	ኤ(e)	እ	ኦ(o)
h	ሀ **ha**	ሁ **hu**	ሂ **hi**	ሄ **he**	ህ **h**	ሆ **ho**		**k/k^{h}**	ከ **ka**	ኩ **ku**	ኪ **ki**	ኬ **ke**	ክ **k**	ኮ **ko**
l	ለ **la**	ሉ **lu**	ሊ **li**	ሌ **le**	ል **l**	ሎ **lo**		**k^{w}**	ኰ **kwa**		ኲ **kwi**	ኴ **kwe**	ኵ **kw**	
m	መ **ma**	ሙ **mu**	ሚ **mi**	ሜ **me**	ም **m**	ሞ **mo**		**ʁ/χ**	ኸ **ġa**	ኹ **ġu**	ኺ **ġi**	ኼ **ġe**	ኽ **ġ**	ኾ **ġo**
r	ረ **ra**	ሩ **ru**	ሪ **ri**	ሬ **re**	ር **r**	ሮ **ro**		**ʁ^{w}/χ^{w}**	ዀ **ġwa**		ዂ **ġwi**	ዄ **ġwe**	ዅ **ġw**	
s	ሰ **sa**	ሱ **su**	ሲ **si**	ሴ **se**	ስ **s**	ሶ **so**		**w**	ወ **wa**	ዉ **wu**	ዊ **wi**	ዌ **we**	ው **wu**	ዎ **wo**
ʃ	ሸ **ša**	ሹ **šu**	ሺ **ši**	ሼ **še**	ሽ **š**	ሾ **šo**		**dz/z**	ዘ **za**	ዙ **zu**	ዚ **zi**	ዜ **ze**	ዝ **z**	ዞ **zo**
q^{h}/q	ቀ **qa**	ቁ **qu**	ቂ **qi**	ቄ **qe**	ቅ **q**	ቆ **qo**		**j**	የ **ya**	ዩ **yu**	ዪ **yi**	ዬ **ye**	ይ **y**	ዮ **yo**
q^{w}	ቈ **qwa**		ቍ **qwi**	ቈ **qwe**	ቍ **qw**			**d/t**	ደ **da**	ዱ **du**	ዲ **di**	ዴ **de**	ድ **d**	ዶ **do**
b/β	በ **ba**	ቡ **bu**	ቢ **bi**	ቤ **be**	ብ **b**	ቦ **bo**		**dʒ~ʒ**	ጀ **ja**	ጁ **ju**	ጂ **ji**	ጄ **je**	ጅ **j**	ጆ **jo**
β	ቨ **va**	ቩ **vu**	ቪ **vi**	ቬ **ve**	ቭ **v**	ቮ **vo**		**g**	ገ **ga**	ጉ **gu**	ጊ **gi**	ጌ **ge**	ግ **g**	ጎ **go**
t^{h}/t	ተ **ta**	ቱ **tu**	ቲ **ti**	ቴ **te**	ት **t**	ቶ **to**		**g/k**	ጐ **gwa**		ጒ **gwi**	ጔ **qwe**	ጕ **qw**	
ʧ	ቸ **ča**	ቹ **ču**	ቺ **či**	ቼ **če**	ች **č**	ቾ **čo**		**ŋ**	ኘ **ńa**	ኙ **ńu**	ኚ **ńi**	ኜ **ńe**	ኝ **ń**	ኞ **ńo**
n	ነ **na**	ኑ **nu**	ኒ **ni**	ኔ **ne**	ን **n**	ኖ **no**		**ŋ^{w}**	ኧ **ńwa**		ኯ **ńwi**	ኰ **ńwe**	ኯ **ńw**	
-	አ **’a**	ኡ **’u**	ኢ **’i**	ኤ **’e**	እ **’**	ኦ **’o**		**ts**	ፀ **tsa**	ፁ **tsu**	ፂ **tsi**	ፄ **tse**	ፅ **ts**	ፆ **tso**
								p^{h}~p	ፐ **pa**	ፑ **pu**	ፒ **pi**	ፔ **pe**	ፕ **p**	ፖ **po**

punctuation marks (ቴርትምልኻትካ) ብሳልሚ/bras ‘()’, ሴዛትኩሚ/full stop/period(።), ኒፂሊባይ/comma (፣), ድርቢባይ/colon (፤), እስቲምልኻት(“”), ካሲውምልኻት/question mark(?), ላኃትኩሚ/preface colon(፡-), ዲብካካሳንኩአኽኼ (…), ዲኒክፂምልኻት(!)

Table 12: Information about the Awngi alphabets and pronunciation

Declaration

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

_____

Wubetu Barud
November, 2016

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

_____

Getachew Mamo