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

Zelalem Mekuria

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

November 2013

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

Zelalem Mekuria

Signature of the Board of Examiners for Approval

Name	Signature
1. Dr.Yaregal Assabie, Advisor	_____
2. Dr. Fekade Getahun, Examiner	_____
3. _____	_____

Dedicated to:

1. **Mekuria G/mariam (My father)**
2. **Yeshihareg Ayele (My mother)**
3. **Selamawit Mekuria (My sister)**

Acknowledgement

Above all, I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully. The idea of doing a thesis on Kafi-noonoo part-of-speech tagger was suggested to me by my father Ato Mekuria G/mariam to whom I wish to thank first and for most to his constructive comments from the beginning up to the end of this thesis. It is also my pleasure to express my heartfelt thanks to my Advisor, Dr Yaregal Assabie, without his invaluable professional scholarly comments; and careful guidance this thesis wouldn't have been possible.

I would like to express my heartfelt thanks to my informants: Ato Simegnih Tekle, Ato Magnecho H/Eyesus and Ato Zelalem Abebe for their limitless cooperation and support.

I am grateful to my mother, Yeshihareg Ayele for her limitless encouragement and support.

The following friends have morally supported and commented the technical aspects of this paper, I would like to acknowledge unreservedly: Yohannes Amde and Kifle Mamo.

Lastly, I wish to thank Addis Ababa University, School of Graduate Studies, for its financial assistance that helped to conduct this study.

Table of Contents

Contents **Page No**

List of Figures

List of Tables

Acronyms and Abbreviations

ANN:	Artificial Neural Network
HMM:	Hidden Markov Model
NLP:	Natural Language Processing
NLTK:	Natural Language Processing Tool Kit
POS:	Part of Speech
POST:	Part of Speech Tagger
KTC:	Kafi-noonoo Temporary Corpus
TEL:	Transformation-based Error-Driven Learning

Abstract

Part-Of-Speech tagger is a program that reads text in given language and assigns parts-of-speech such as noun, verb, adjective, etc. to each word and other token within the text. Several part-of-speech taggers are available on the web for different languages including Amharic, Oromifa and Tigrigna. However, these POS taggers cannot be applied directly for Kafi-noonoo language. Thus, this thesis presents a research work on Kafi-noonoo part-of-speech tagger. In order to develop the tagger, the study employed a hybrid approach i.e. HMM and rule-based tagger at sentence level. Developing part-of-speech tagger for a language has many advantages such as: it can be used as input for full parser; it can be used in text-to-speech system to correct the way of pronunciation, it can be used for surface linguistic analysis, it can be used as a pre-processing step for researchers who want to conduct higher level NLP application development and it also provide a way of learning the language by discovering the word category and grammar construction of the language.

For training and testing purpose, 354 untagged Kafi-noonoo sentences are collected from two genres and annotated using an incremental corpus preparation approach. In addition to this, 34 part-of-speech tags are identified for tagging purpose. 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 probability for the statistical component of the hybrid tagger and set of transformation rules for the rule-based component of the hybrid tagger. Based on these probabilities and transformation rules, the hybrid tagger (combination of HMM and rule-based tagger) assigns the most suitable word class information for the given untagged Kafi-noonoo texts. The performance of the prototypes i.e. HMM, rule-based and hybrid taggers are tested using different experiments. As a result, HMM and rule-based tagger with unigram initial state tagger shows 77.19% and 61.88%accuracy respectively whereas, the hybrid tagger improve the accuracy to 80.47%.

Key words: Part of speech tagger, HMM, Rule-based, Hybrid tagger and Transformation rules

CHATPTER ONE

Introduction

1.1 Background

Natural language processing (NLP) is a field of computer science and linguistics that is concerned with the interactions between computers and human (natural) languages. It began as a branch of artificial intelligence [4]. In theory, natural language processing is a very attractive method of human computer interaction. It is normally used to describe the function of computer system which analyzes or synthesizes spoken or written language [14]. One of the challenges inherent in natural language processing is teaching computers to understand the way humans learn and use languages [30]. There are various research attempts under investigation; some of these include machine translation, information extraction and retrieval using natural language, text-to-speech synthesis, automatic written text recognition, grammar checking, and part-of-speech (POS) tagging [10]. Most of these NLP applications have been developed for languages like English [6, 7], Turkish [19] and Ethiopian language including Amharic [28], Afaan Oromo [10], and Tigrigna [29].

POS tagging is the act of assigning each word in the sentence a tag that describes how that word is used in the sentence. That means it assigns whether a given word is used as a noun, adjective, verb, etc. Most tagging algorithm fall in to one of two classes [5]: rule-based and statistical taggers.

Rule-based taggers use hand coded rules to determine the lexical categories of a word [15]. Words are tagged based on the contextual information around a word that is going to be tagged. Part-of-speech distribution and statistics for each word can be derived from annotated corpora dictionaries. The dictionary provides a list of words with their lexical meanings. In the dictionaries there are many citations that describe a word in different context. These contextual citations provide information that is used as a clue to develop a rule and determine lexical categories of the word [10]. On the other hand, statistical methods assign tag for a word by calculating the most likely tag in the context of the word and its immediate neighbor [26]. The main idea behind all statistical tagger is a simple generalization and picks the most-likely tag for

this word [10]. A statistical approach includes most frequent tag, n-gram and hidden markov model (HMM) [10].

Nowadays part-of-speech tagger is developed for different languages and it remains an intensive research area for other different languages. As to the best of the researcher‟s knowledge, Kafi-noonoo is a language which does not have any POS tagger developed so far.

Kafi-noonoo is a language spoken in south western Ethiopia by Kafecho people whose population is estimated to be 3 million [9]. It belongs to the Afro Asiatic language super family of the North-omotic Southern Gonga Sub-group. In the past 20 years much emphasis was not given for the language. But since 1987 the language is offered as independent course both at primary and secondary level with in the zone.

1.2 Statement of the Problem

Over the past few years, several natural language processing applications such as machine translation, speech-recognition, part-of-speech tagging etc. have been developed for local languages. Among these applications, automatic part-of-speech tagging is a useful form of linguistic analysis that contributes a lot in the effort of NLP applications development. It used as a pre-processing component for sentence grammar checker, spell checker, information retrieval, etc.

Despite its importance, there is no research conducted on POS tagger development which is becoming an obstacle for research and development works on higher level NLP applications. In fact there are POS taggers that have been developed for local language like Amharic, Afaan Oromo and Tigrigna. However, these POS taggers cannot be applied directly for Kafi-noonoo language. Thus, developing automatic part-of-speech tagger for Kafi-noonoo language contributes a lot in the field of natural language processing application.

1.3 Objective of the Study

1.3.1 General Objective

The general objective of this research is to design and develop a POS tagger for Kafi-noonoo language.

1.3.2 Specific Objective

The specific objectives of this research are to:

- identify and review techniques for POS tagging
- identify word category and tagset for Kafi-noonoo language
- study the structure of Kafi-noonoo sentence
- collect and prepare corpus for training and testing purpose
- design and develop a POS tagger for Kafi-noonoo language
- develop Kafi-noonoo POS tagger prototype
- test the performance of the prototype

1.4 Methodology

1.4.1 Literature Review

In order to understand the current state of the art in the area of automatic part-of-speech tagging, different literatures that are relevant to this thesis will be reviewed.

1.4.2 Data Collection

In this thesis, we use two different datasets. The first dataset from Kaffa Cultural and Tourism Bureau is composed of 1000 proverbs. The second dataset is 5 long reading passages taken from Grade 9 and 10 Kafi-noonoo student books. In order to tag the raw text with their corresponding word classes, we used incremental tagging approach.

1.4.3 Development Tools

The following tools used in the experiment to develop prototype of the model

- NLTK (open source Natural Language Processing Tool Kit)[3, 27] version 2.0.4

- Python programming language[24, 27] version 2.6.6

1.4.4 Modeling

In this thesis, three different models constructed namely HMM based tagger, rule-based tagger and hybrid tagger (combination of HMM and rule-based tagger).

1.4.5 Evaluation

After building the model for HMM, rule based, and hybrid tagger (combination of HMM and rule based tagger), the performance of the system was evaluated using different experiments.

1.5 Scope and Limitation of the Study

Tags are labels that provide different information like word category, gender, number and tense. Tagsets are the collection of tags that can be used in different NLP applications. Due to language features, objective and purpose, the tagset developed for one language can‟t be used for another language.

Corpus (plural corpora), is a collection of text with or without additional linguistic information. Corpora can be single or multiple categories.

The raw texts collected from one or different genre like newspaper, fiction, text book, reports, scientific paper, etc. are tagged with their corresponding word class based on the tagsets. However, we were not able to find such documents (tagsets and corpora) written in Kafi-noonoo language. Thus, this work is subject to the following scope and limitation:

- The tagset provide only word class information
- The corpus is prepared from two genres that is text book and proverbs

1.6 Application of the Study

POS tagging is a useful form of linguistic analysis. There are many applications of Kafi-noonoo POS tagger, some of which are presented as follows:

- It can be used as an input for full parser
- It can be used in text-to-speech system to correct the way of pronunciation

- It can be used for surface linguistic analysis
- It can be used as a pre-processing step for researchers who want to conduct higher level NLP application such as spelling checker, grammar checker, question answering, etc.
- It provides a mechanism to learn Kafi-noonoo as a second language, by discovering the word category and grammar construction mechanism.

1.7 Thesis Organization

This thesis is organized as follows. Chapter 2 presents literature review and different related works on POS tagger. Chapter 3 focuses on the study and assessment of the nature and structure of Kafi-noonoo sentences, word classes and tagset preparation. Chapter 4 presents the design of POS tagger for Kafi-noonoo language. Test result where shown and discussed in Chapter 5. Finally, in Chapter 6, conclusion and future works are presented.

CHAPTER TWO

Literature Review and Related works

2.1 Introduction

As it has been discussed in Chapter One, part-of-speech tagging (POST), also called grammatical tagging or word category disambiguation, is the process of labeling a word in a text 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 [5, 27]. It is an important component of natural language processing applications and plays an important role in speech synthesis, speech recognition, information retrieval, word sense disambiguation, etc.

There are many publicly available POS tagger on the web for different languages. For example it is possible to see the English version of Hidden Markov Model (HMM) based part-of-speech tagger using NLTK. Given the text „The boy has gone"; it generates word class information that is shown in Figure 2.1.

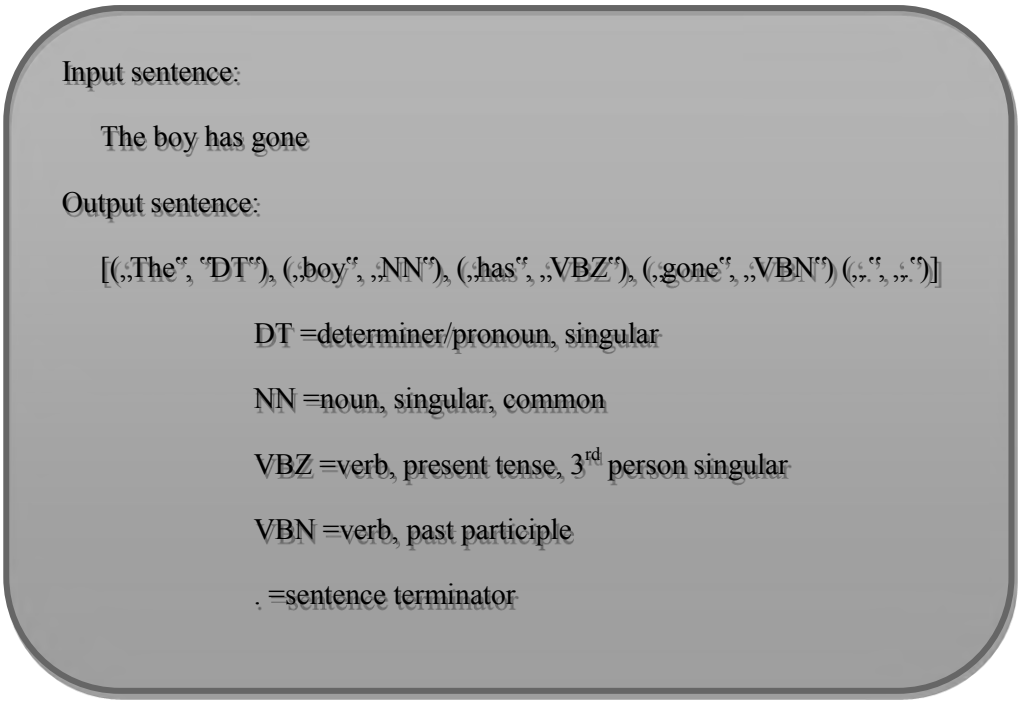

Figure 2.1 Example of HMM based part-of-speech tagger using NLTK

As we can see from Figure 2.1, the input to the tagger is a string. Except the last token i.e. period (.), the output of the tagger is a single best tag that provides word class, number and tense information about each token within the sentence.

2.2 Approaches to POS Tagging

Several approaches have been proposed to annotate words automatically with their part-of-speech tags. The most common ones are rule-based [6, 7, 8, 20 and 21], stochastic [10, 16], artificial neural network [12] and hybrid approaches [16, 17, 19, 28 and 29]. Rule-based taggers [6, 7, 8, 20 and 21] assign a tag to each word using a predefined set of constraints, contextual information or automatically generated rules from the training set. The stochastic (probabilistic or statistical) approach [10, 16] uses frequency, probability or statistical information to assign the most appropriate tag sequence for a given sequence of words. Artificial Neural Network (ANN) [12] tries to learn pattern from the training data and sets network weight as required for tagging or other pattern classification problem. The hybrid approach may combine either the benefit of rule-based and probabilistic approach or rule-based and artificial neural network approach [16, 17, 19, 28 and 29].

2.2.1 Rule Based Approach

Rule based approach uses rules to assign tag to words. According to [29], the rules depend on linguistic features of specific language such as morphological, lexical and syntactical information. These rules may be developed by a linguistic professional or by machine learning on a pre-tagged corpus [6, 7, and 20]. The first way of getting rules is difficult, time consuming and prone to error. In addition to these, it requires language expert on the specific language being tagged. While in the second case a model tries to learn and store sequence of rules (transformation rules) using a training data without manual rule construction [7, 20]. This way of obtaining rule is called Brill transformation based approach and explained in the work of [7]. Other than hand written and machine learned rules, the rule based approach uses a contextual information rules. These rules are often known as context frame rules [8, 19]. For example: a context frame rule for English might be as follows: if an ambiguous/unknown word X is preceded by a determiner and followed by a noun tag it is an adjective. Other than this most

taggers use morphological information in order to support the tagging process [19]. For instance: if an ambiguous word ends in an **–ous,** label it as adjective.

Generally the rule based approach that involves manual rule construction is laborious, time consuming, prone to error and require knowledge of the language being tagged. According to [6], rule based tagger has many advantages. These are: less stored information, small set of simple rule, ease of finding and implementing improvements to tagger, portability from one tag set, corpus or language to another.

2.2.2 Stochastic Approach

Most current part-of-speech taggers are stochastic. This approach uses probabilistic model to assign the most suitable tag for a given word by calculating the most appropriate tag based on the word and its neighboring tag. Based on the training data it uses, this approach can be classified as supervised and unsupervised tagger. Supervised taggers uses corpus where word class is attached to every token [16] which is used for training to learn information about tagset and word frequencies. On the other hand, unsupervised tagger uses a Baum-Welch algorithm to learn information about tagset and word frequency from untagged corpus [8, 16]. This algorithm is used when no pre-tagged corpus exist for training purpose. This approach can be based on different methods n-gram, maximum-likelihood, and HMM [8].

Maximum-Likelihood (Most Frequent Tag) Method

It is the simplest method under the stochastic approach. It assigns the most frequent part-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:

$$P(w/t) = \frac{Count(w,t)}{Count(t)} \quad (2.1)$$

Where w and t are word and tag respectively.

The major problem of this method is it does not consider local contextual information to assign the most appropriate tag for a given word. It rather picks the most frequent tag for a given word.

N-gram Method

The N-gram method is a way of dealing contextual information of a word within a given sentence. It uses the frequency of part-of-speech, word or both word and corresponding part-of-speech sequence to produce the probability of a word or other class string by considering contextual information within a given sentence. According to [5], it is originally created as a way of estimating the next element (word, tag or tag of current word) of sequence given only the previous n-1 elements. It can be used to calculate the probability of tag sequence that is the probability of t_i given the previous n-1 tags. This can be represented mathematically as:

$$P(t_i/t_{i-1}, t_{i-2}, \dots, t_n) \quad (2.2)$$

It can be used to calculate the probability of word sequence i.e. the probability of w_i given the previous n-1 words. This can be represented mathematically as:

$$P(w_i/w_{i-1}, w_{i-2}, \dots, w_n) \quad (2.3)$$

Other than calculating the next tag and word given only the previous n-1 tag and word respectively, the n-gram method used for calculating tag of current word given the previous n-1 words. This can be represented mathematically as:

$$P(t_i w_i/w_{i-1}, w_{i-2}, \dots, w_n) \quad (2.4)$$

Changing the value of n will generate a different n-gram method. In part-of-speech tagging most of the time the value of n is equal to 2 or 3 and the resulting methods are called bigram and trigram respectively. According to [16] the value of n is dependent on the training data.

HMM Method

Hidden Markov Model (HMM) 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.

According to [5] the parameters needed to define HMM are:

- States: a set of state
- Observation sequence: a set of legal accepting states
- Transition probability: probability distribution that govern the transition from one state to another state.
- Observation likelihoods: a set of observation likelihoods
- Initial distribution: an initial distribution over states.

More formally it can be represented as a set{S, O, A, B, Π} [18].

Where

- S is a set of N states where individual states denoted by S= {S_1, S2…, SN}, state at time t as q_t
- O is a set of T observation sequence where each observation is represented by O= {O_1 O_2 …O_T}
- A=a_{11} a_{12}…a_{nn} is a transition probability matrix where each a_{ij} represents a probability of transition from state i at time t to state j at time t+1. The sum of all transitions from state i to state j is 1. It can be defined as $a_{ij}=P(q_{t+1}=S_j|q_t=S_i)$ where 1<=i, j<= N , a_{ij}>=0 and (The sum of all transitions from state i to state j is 1).
- B is the emission or observation probability distribution of each state. Emission or observation probability $b_j(k)$ can be computed as $b_j(k)=P(O_i=k|S_i=j)$ for 1<=j<=N and 1 <= j <= M where M is the number of distinct observation symbol per state.
- Π (initial distribution) is an initial probability distribution over state, such that Π is the probability that HMM will start in state i.

In general a complete specification of HMM can be represented as a set containing {S, O, λ} [18] where

- S is a set of states
- O is a set of observation
- λ= {A, B, Π} is a set that contain probability measure of A, B and Π

Applying HMM in POS tagger development consists of two main tasks: the first task is estimating the model parameters A, B and Π from the training set and the second one is computing the most likely sequence of the original state transition given new observation. When the HMM method is taken to the application of POS tagging, the hidden states are POS tags and the sequence of words are the sequence of observations. The transition probability in POS tagging is the probability of moving from one tag to the next and the emission probability is the probability of getting a word W_i being tagged as T_i. If we have a sequence of words w_1, w_2…, w_n, the goal of HMM tagger is to select the most likely tags t_1,t_2…, t_n associated with those words.

According to [5] HMM taggers try to find the tag sequence that maximizes the following formula:

$$P(\text{word/tag}) * P(\text{tag/previous n tags}) \quad (2.5)$$

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).

P(tag/previous n tags) is the probabilitythat one tag follows another (N-gram)

In general, stochastic based POS taggers have advantages and disadvantages. The advantages and disadvantages of these taggers are summarized in Table 2.1.

Table 2.1 Advantages and disadvantages of stochastic based tagger

Stochastic based tagger	
Advantage	Disadvantage
• It may not need linguistic expert • The performance of the tagger depend on the amount of data from different source	• Not suitable for language with less annotated data • It has huge amount of stored information

2.2.3 ANN Approach

Artificial neural network is a system that is composed of many simple processing elements operating in parallel whose function is determined by network structure, connections strength, and the processing performance at computing element or node.

The other definition of artificial neural network is given by [11] and defines it as a massively parallel distributed processor that has a natural way for storing learned knowledge and making it available for use.

It resembles the human brain in two aspects:

1. Knowledge is acquired by the network through learning process.
2. Inter-neuron connection strength known as synaptic weights are used to store knowledge.

The key element of this paradigm is the new structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems [25]. Learning in biological system and ANN involve adjustment of the synaptic connection that exists between neurons.

The most common types of artificial neural network are arranged vertically in three layers [12] these are input, hidden and output layer.

- Input layer represent the original information that is fed into the network and it is connected to the hidden layer.

- Hidden layer is the one that is connected with the output layer. It‟s activity determined by the activity of the input layer and the weight on the connection between the input and hidden layer.
- Output layer represent the outcome of the learning process from input and hidden layer. Its‟ behavior depend on activity of hidden units and the weight on the connection between the hidden and the output layer.

When ANN approach is taken in to POS tagger developments task, according to [29] before working on the actual ANN based tagger, it requires a pre-processing activity. The output of the pre-processing activity‟s 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 is produced.

In general, ANN based taggers have advantages and disadvantages. The advantages and disadvantages of these taggers are summarized in Table 2.2.

Table 2.2 Advantages and disadvantages of ANN based taggers

ANN based tagger	
Advantages	Disadvantages
• It is suitable for languages having small number of tagset and small amount of training corpus • It combines the advantage of HMM and trigram tagger	• As the number of tagset increase, the performance of the tagger become worse • It has lower processing speed compared to stochastic approach • Both selection and treatment of ambiguous words are performed by only considering the corpus

2.2.4 Hybrid Approach

This approach combines either rule-based and probabilistic approach or rule-based and artificial neural network approach to enhance the performance of the part-of-speech tagger. Some researchers used hybrid approach (stochastic and rule-based approach) [17, 19, and 29]. Others implement hybrid approach that combine artificial neural network and rule-based approach [28].

2.3 Related Works

2.3.1 Previous Work on Local Languages

The first Afaan Oromo language part-of-speech tagger using HMM approach was developed by Getachew Mamo and Million Meshesha [10]. In this work, the researchers used HMM approach which is one of the most common mechanisms under stochastic approach. For training and testing purpose, the researchers collected 159 sentences (with a total of 1621 words) from different sources to make the corpus balanced and they used 17 tagset.

In the tagging process, the tagger assigns word classes to a given Afaan Oromo text with two main phases. In the first phase, the tagger trains on the training data in order to compute and store both lexical and transitional probability of training data. In the second phase, the tagger accepts untagged Afaan Oromo text and tokenized into words. Then, the tagger assigns the correct POS tag for each token. This is achieved by using unigram and bigram model of the Viterbi algorithm by taking the stored information during the first phase.

The researchers have tested the performance of the tagger using tenfold cross validation mechanism. As a result they have got 87.58% and 91.97% accuracy for unigram and bigram model respectively.

Another work on Afaan Oromo language part-of-speech tagger was done by Mohamed Hussen [21]. In this work, the researcher adopted transformation based tagging (Brill tagging), with some modification on the original tagging template. This approach is an instance of transformation-based learning (TBL) approach to machine learning [6], and it draws inspiration from both the rule based tagger and stochastic taggers. Like the rule based taggers, it is based on rules that specify which tag should be assigned to a word. But like stochastic taggers, it is a machine learning technique in which rules are automatically induced from the training data. The researcher used 233 sentences (1708 words) from different source to make the corpus balanced. Out of this, 90% of the corpus used for training purpose while the remaining 10% used for testing. In addition to this he has identified 18 tagset.

In order to learn a set of transformation rules, TEL (Transformation-based Error Driven Learning) based tagger takes unannotated corpus as input and goes through the initial-state

tagger, that assigns the most likely tag. This gives a temporary output. The temporary corpus compared with the one that is manually tagged and assumed to be correct. Based on the result of comparison, the temporary outputs pass through both lexical and contextual learners to drive rule of transformations. Each transformation rule is tasted on the temporary output to select the best transformation rule that maximize the performance of the tagger. Rule with best score applied to the temporary corpus in order to produce the next temporary corpus and added to the set of rule. Starting from the comparison of temporary corpus with reference text to transformation rule generation, the process continue in the same fashion to produce all the permissible rule with the corresponding temporary text until no rule can further improve the tagging of the temporary corpus. Afterwards, the tagger accepts unannotated Afaan Oromo text and assigns the best tag for each word based on the rules that is learned.

To test the performance of the proposed method and original Brill tagger, the researcher has conducted ten experiments on different number of words. As a result he has got an average accuracy of 80.80% and 77.64% for modified and original Brill tagger respectively.

A hybrid approach by applying a combination of rule-based and statistical approaches has been introduced for Tigrigna language part-of-speech tagger by Teklay G/Egzabiher [29]. In this work the researcher used a combination of HMM, which is widely used under stochastic approach and adapting Brill transformation-error driven learning approach to drive machine learned rules for designing the rube based tagger component.

The researcher has collected a total of 26,000 words from Tigrigna news broadcasting agencies and annotate manually with their corresponding word class. In addition to this he has identified 36 tagsets for the entire tagging process. Among the total word, 75% (20,000) words used for training purpose while the remaining 25% (6000) words used for testing purpose.

Generally this study finds tag of a word in two main steps. The first step is performed by the HMM tagger. The HMM tagger first annotates the given raw text and provides a level of confidence (threshold value) for each tag sequences. In the second step, the confidence level of each tag sequence compared with the minimum confidence level that is set by the researcher using the output analyzer module. 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.

In order to test the accuracy of the proposed method, the researcher conducted different experiment for the three types of taggers namely HMM tagger, rule based tagger and hybrid tagger. As a result he has got an accuracy of 89.13% for HMM, 91.8% for rule based and 95.88% for hybrid tagger.

2.3.2 Previous Work on Foreign Languages

Eric Brill [6, 7] proposed another approach to POS tagger called Brill tagger. It is a kind of transformation based learning named after its inventor. He proposed a system that guesses the tag of each word, then goes back and fix the mistakes. The general idea of the tagger is to assign each word within a given text it‟s most likely tag estimated by initial-state tagger that is trained on a large tagged corpus without regard to context [6]. Once the text passed through the initial state tagger, it compared with the reference text (manually tagged text). As a result, an ordered list of transformation rules are learned that can be applied to the output of the initial-state tagger to make it better resemble with the reference text.

In order to test the performance of the system, the researcher conducted an experiment using 1.1 million words of the Pen Tree-bank tagged Wall Street Journal corpus. From these, 950,000 words where used for training and 150,000 words where used for testing. Out of 950,000 words of the training corpus, 350,000 words where used to learn rules for tagging unknown words and 600,000 words where used to learn contextual rules. As a result, the experimental performance of this approach indicates 96.6% accuracy on the test corpus.

A composite POS tagger for Turkish, which combines rule based and statistical approach, was developed by Levent Altanyart, Zihni Orhan and Tunga Gunger [19]. In this work, the researchers used both word frequencies and n-gram (unigram, bigram and trigram) probabilities. In order to increase the performance of the system, the researchers use two additional features. In the first case, they incorporate a morphological analyzer which is used to obtain the part-of-speech of words independent of the words with in the corpora. This enables the system to guess the tag of the word even if it does not exist within the corpus. In the second case, the researchers

use another statistical approach which is related to the part-of-speech of words based on the position within the sentence.

In order to develop the system, the researchers have collected 7200 sentences. Among the 7200 sentence, 6000 (85%) of the corpus used for training and the reaming 1200 (15%) of the sentences are used to testing purpose. In addition to this, they identified 13 word classes for the tagging process.

The system finds the tag of a word in three main steps. In the first step, the statistical analyzer module computes some statistical data from the training corpus. In the second step, the tag set finder find possible part-of-speech for words to be tagged. Finally, the main modules of the system determine part-of-speech of words. To reach final decision the tagger combines word frequencies, n-gram probabilities, heuristics data and data about candidate tags.

To test the performance of the system, with and without statistical data, the researchers have performed three experiments by using different part of the corpus as training and testing set. As a result they have got an average accuracy of 82.26% for system with statistical data and 66.73% for system without statistical data.

The summarized version of each work is shown in Table 2.3.

Table 2.3 Summary of related works

Solution	Objective	Methodology	Drawbacks
Part of Speech Tagging for Afaan Oromo[10]	• Design Afaan Oromo language part of speech tagger.	• Use HMM approach • It uses 10-fold cross validation mechanism.	• Maintain large statistical tables
Part of Speech Tagger for Afaan Oromo Language Using Transformation-Based Error-Driven Learning [21]	• Design rule-based part of speech tagger for Afaan Oromo language.	• Use Transformation-Based Error-Driven Learning approach. • For testing purpose it use K-fold cross validation mechanism.	• Some templates work only for this Tagger (Example-template that consider prefix)
Part of Speech Tagger for Tigrigna Language [30]	• Design hybrid part of speech tagger for Tigrigna language.	• Use Hidden Markov Model and Transformation-based Error-driven Learning approach.	• Threshold value is computed only for words not for entire sentences. • Use less window size i.e. two (bi-gram)
A Simple Rule-Based Part of Speech Tagger [6, 7]	• Design Transformation- based error-driven learning tagger for English language.	• Use TEL approach • Use machine learned rules	• Some templates work only for this Tagger (Example-length of deleted word to form new words)
A Composite Approach for Part of Speech Tagging in Turkish [19]	• Design a composite part of speech tagger for Turkish language.	• Use rule-based and statistical approach • It also use morphological and words position features.	• The word position features not functional for words other than initial and final position.

CHAPTER THREE

Linguistic Properties of Kafi-noonoo Language

3.1 Introduction

As we have seen in Chapter One, Section 1.1, the language of Keficho is known as Kafi-noonoo, which literally means the mouth of Kaffa [9]. It belongs to the Afro-Asiatic language super family of the North-omotic Southern Gonga Sub-group. In this sub group, Sheka, Enarya and Garo (Bosha) are also included. Although Kafi-noonoo was unwritten language for a long time, it has recently started using the Latin Script for writing purpose and the language is being used as a medium of instruction at the elementary education level. In some institutions like Bonga Teachers training institution, all the subjects are taught in Kafi-noonoo for elementary teachers. It also thought as a subject in the junior and secondary schools of Bonga.

Kafi-noonoo has 22 consonant phonemes. Out of these, six of them are both long and short consonants. Among the 22 consonants, five of them are borrowed from English and Amharic languages. Table 3.1 shows a list of consonants and borrowed letters in Kafi-noonoo language.

Table3.1 Consonants and borrowed letter in Kafi-noonoo language

Consonant (Shemmeebeetinaʺo)	Short consonant	Example	Long consonant	Example	Borrowed letter (TawusheBireenaʺo)
b, c, f, g, h, j, k, l, p, q, r, s, t, v,w, x, y, z,ch,sh, ts, zh	b	Gabo	bb	Gabbo	v, z, ny, ts, zh
	d	Yudo	dd	Yuddo	
	g	Shago	gg	Shaggo	
	m	Timo	mm	Timmo	
	n	Gino	nn	Ginno	
	ph	Kepho	pph	Keppho	

In addition to the consonants, it has five long and short vowels. Table 3.2 shows list of vowels in the Kafi-noonoo language.

Table 3.2 List of short and long vowels

Short vowel	Example	Long vowel	Example
a	Baro	aa	Baaro
e	Kexo	ee	Keexo
i	Gino	ii	Giino
o	Qoco	oo	Qooco
u	Shuno	uu	Shuuno

The long vowels and consonants can be obtained by doubling the corresponding short vowels and consonants respectively. The difference in length of both vowels and consonants induces difference in meaning. For example the word „baro" means „corn" while „baaro" is „forehead".

In Kafi-noonoo tone has a semantic and grammatical function. For example:

Kemo „buy" high and low tones

Kemo „sell" both high tones

In general Kafi-noonoo has 32 letters including letters borrowed from other languages. Out of the 32 letters, by excluding browed letters, 27 of them treated as Kafi-noonoo alphabet. Figure 3.1 shows Kafi-noonoo alphabet.

A B C D E F G H I J K L M N O P Q R S T U W X Y CH PH SH

Figure 3.1 Kafi-noonoo alphabets

3.2 Kafi-noonoo Sentence Structure

Like other languages, in Kafi-noonoo a sentence is a set of words that contain:

1. A subject (the topic of the sentence)
2. A predicate (what is said about the subject)

For example: - Mesfini hammite (Mesfen has gone).

Mesfini = subject hammite = predicate

Kafi-noonoo sentence can be classified as simple (xolle) and compound (xappe) sentence. Simple sentences have only one verb. For example: Bushoo **hammite** (The boy has **gone**)

In case of compound/complex sentence, the sentences have more than one verb. For example: Ta Bunno **guppemmona** aree kexo **hidiyan** (When I **prepare** coffee, she **cleaned** the house)

3.3 Word Classification

There are thousands of words in any language. But not all words have the same job. For example, some words express action. Other words express a thing. Other words join one word to another word. These are the building blocks of the language. When we want to build a sentence, we use the different types of word. Each type of word has its own job.

In order to determine the category of the word, linguists use morphological (internal structure of a word), syntactic (structure of the sentence) and semantic (meaning of a word) as a clue [27].

3.3.1 Kafi-noonoo Word Classes

Words in Kafi-noonoo can be divided into two broad categories: closed class types and open class types [2]. Closed classes are those that have relatively fixed members. While open classes are those continually changed or borrowed from other languages.

According to [2, 1], Kafi-noonoo has six word classes: noun, adjective, verb, adverb, preposition and pronoun. Some of these are divided into other sub-classes. For example, pronoun class is categorized as personal pronoun, demonstrative and interrogative pronoun. A detailed description of Kafi-noonoo word classes were used by [1 and 2] is given in the subsections.

3.3.1.1 Noun (Shigo)

Noun is a part of speech that names a person, place, thing, quality, quantity or concepts. Kafi-noonoo nouns, like English nouns, are words used to name or identify a person, place or things. For example:

- Persons: Asho (Person), Mesfini (Mesfen), Wondimi (Wondemu)
- Place: Boonga (Bonga)
- Things: Kexo (house), mixo (wood)

It has four sub-classes. Table 3.3 shows sub-class of noun in Kafi-noonoo language.

Table 3.3 Sub-classes of noun

No	Sub-classes	Example
1	Common noun (Gogeeshigo)	Mixo, Kexo, Asho
2	Abstract noun (Abstrakteshigo)	Nallow, Shalligoo, Emiroo
3	Collective noun (Kiceeshigo)	Maccoo, Yobbero, Maachoo
4	Proper noun (Qellishigo)	Boonga, Qocciti, Juppiteero

In Kafi-noonoo there are few word endings that show a word is a noun, For example:

- -na‟o > Bushiishee**na'o** / Noun = children‟s
- -ch > Bushoo**ch** / Noun = for childrens

But this is not always true for the word endings of **–na'o**, **-ch** and others. For example:

- -na‟o > Accechi**na'o** / Adjective = wise
- -na‟o > Hammiti**na'o** / Verb = went

As we can see form the above Example, all words are ended with –na‟o endings but they represent different word classes.

3.3.1.2 Verb (Kanno)

Kafi-noonoo has a subject, object, verb (SOV) word order. The verb is a word that describes the subject‟s action or state within a sentence. For example:

- Garkalli yiich chatt **hammite.** (Gebremichaeal went to chetta yesterday)

- Wondimi bi kechishuunoon **shuuneebeete.** (Wondemu is doing his homework)

3.3.1.3 Adjective (Ashigeena'o)

An adjective is a word that describes, identify or quantify a word by preceding the noun or pronoun which it modifies. For example: naccoo (white), cello (red), aaʺo (black), gissho (small), oogoo (large), etc.

- **Nacce** qoreddoo (white close)
- **Ooge** kexo (large house)

In the above sentence **naccoo** and **oogoo** are adjectives that describe the noun **qoreddoo** and **kexo** respectively.

3.3.1.4 Pronoun (Shikeroo)

In Kafi-noonoo, pronouns are small words that take the place of a noun. We can use a pronoun instead of a noun. Pronouns are words like: ne (you), bi (he/she), no (we), etc. it has three sub-classes. Tale 3.4 shows sub-class of pronoun.

Table 3.34 Sub-classes of pronoun

No	Sub-class	Example
1	Personal pronoun (ashinneeshikeroo)	ta, ne, bi, no, boonoshi, ittoshi
2	Demonstrative pronoun (bekkiyeeshikeroo)	ebi, okkebi, hini, meni, meno
3	Interrogative pronoun (echee Shikeroo)	koni, kooch, konich, amo, aabi

3.3.1.5 Preposition (Shigillo)

In Kafi-noonoo prepositions are words usually coming in front of, a noun or pronoun and express source, destination, location and relation to another word or element. For example:

- Ta **gubb** (after me)
- Ne **aff** (before you)
- Qeechi**dech** (under the bed)

3.3.1.6 Adverb (Shakanno)

An adverb is a word that tells us more about a verb. It qualifies or modifies a verb, adjective and other adverbs. In Kafi-noonoo modifiers of verb or verb phrase usually express time, location, manner, etc. For example:

- Qocciti **yiich** wate. (kochito came **yesterday**)
- Qocciti **kaatee** waate. (Kochito came **quickly**)

In the first example, the word **yiich** (yesterday) used as adverb of time while in the second sentence, the word **kaatee** (quickly) used as adverb of manner.

3.4 Kafi-noonoo Tagsets

The previous section gave broad description of word classes that Kafi-noonoo words fall into. In this section, the actual tags used in this thesis are discussed.

As far as the researchers‟ knowledge is concerned, there is no publicly available tagset for Kafi-noonoo language. In order to identify and develop tagset for this thesis, the researcher has made interview and continuous discussion with Kafi-noonoo language professional Ato Simegnih Tekle, Ato Magnecho H/Eyesus and Ato Zelalem Abebe who have knowledge of the language. Ato Simegnih Tekle, Ato Magnecho H/Eyesus and Ato Zelalem Abebe are Kafi-noonoo teachers in Bonga College of Teacher Education.

The tagset that are discussed below are classified as basic classes and sub-classes of the basic classes are noun, verb, adjective, pronoun, adverb, preposition are considered. In addition to these, conjunction, interjections, numerals and punctuations are also included as basic classes of Kafi-noonoo language. The hierarchical structure of the tagsets used in this thesis is shown in Figure 3.2.

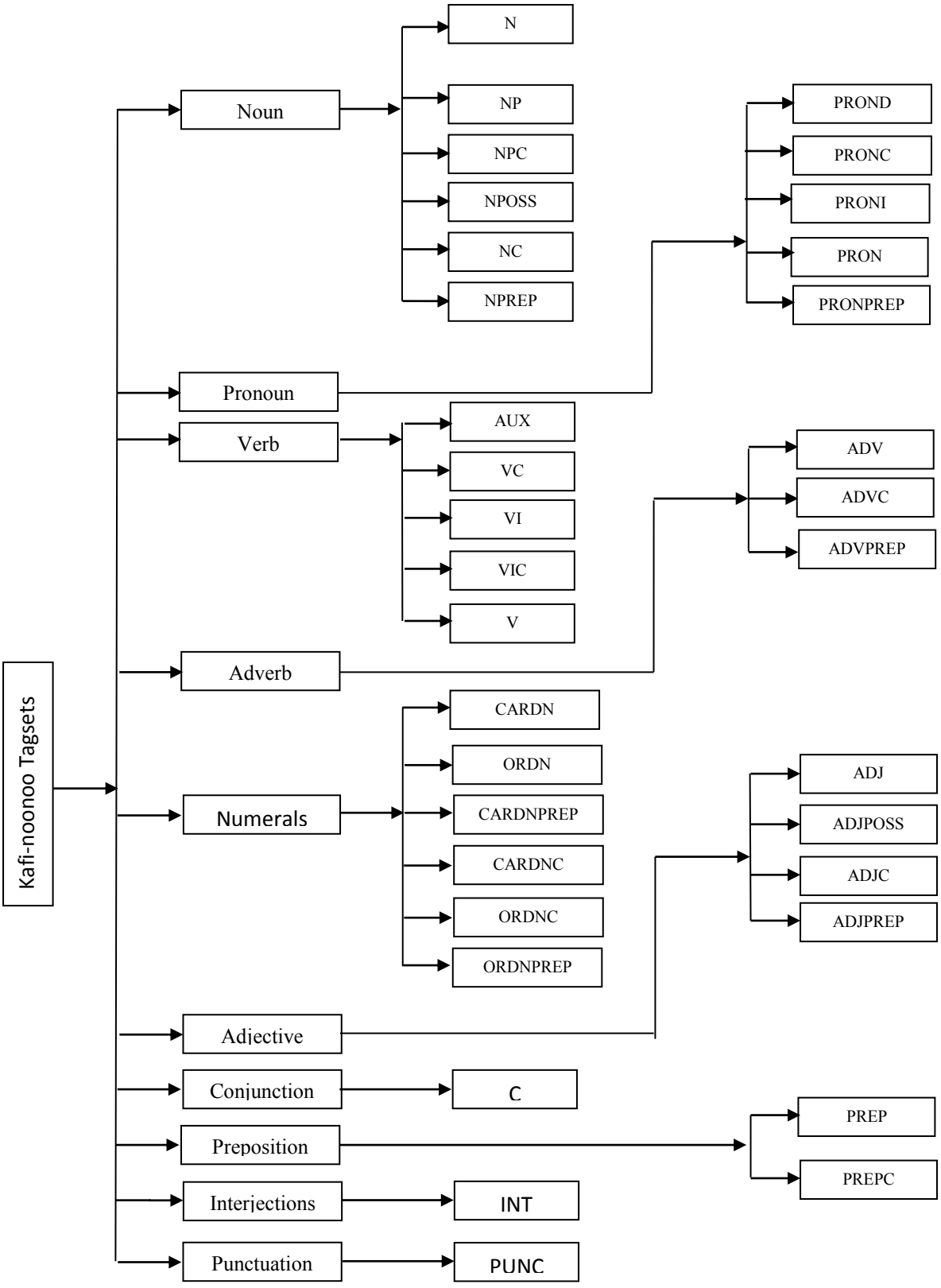

Figure 3.2 Hierarchical structure of Kafi-noonoo tagset

Noun and its sub-classes

In Kafi-noonoo nouns have different attributes like numbers, genders and definiteness which can be common noun (Gogeeshigo), abstract noun (abstrakteshigo), collective noun (Kiceeshigo) and proper noun (qellishigo). Due to tagset complexity problem, we did not include the entire attribute except for proper noun. In this main class we identify noun as a general class and proper noun, proper noun with conjunction, noun conjunction, noun preposition, and noun possession as a sub-classes. This class and its sub-classes are explained in the following examples.

- Nouns that represent the name of a person, place, thing, organization, etc. are considered to be proper noun and tagged by NP. For example: **Boonga** (Bonga), **Garakaalli** (Gebremichael)
- Proper noun can be attached with conjunction. This type of proper noun classified under proper noun with conjunction sub-class and tagged by NPC. For example: yeeri**n**
- Nouns affixed with conjunction are considered as noun conjunction and tagged by NC. For example: Abbebi**naa** Jemallina (Abebe and Jemal)
- Nouns suffixed with preposition are considered to be noun preposition sub-class and tagged as NPERP. For example: kaamelo**ona** (by car)
- Noun can be attached with possession. This type of noun classified under Noun possession and tagged by NPOSS. For example: ashich
- Other forms of nouns that cannot be classified under the above classes such as collective noun (kiceshigo), abstract noun (abstrakteshigo), and common noun (gogeeshigo) are tagged by N. For example: **mixo** (wood/common noun), **maccoo** (people/collective noun), **shano** (love/abstract noun)

Pronoun and its sub-classes

In this thesis, we identify pronoun as a general class and demonstrative pronoun, pronoun conjunction, interrogative pronoun, and pronoun preposition as its sub-classes. This class and its sub-classes are explained using the following examples.

- Pronouns that point or identify noun or pronoun classified under demonstrative pronoun and tagged by PROND. For example: **ebi** (this), **menoshi** (those), **okkebi** (that)
- Pronouns attached with conjunction classified under pronoun conjunction and tagged as PRONC. For example: biin**naa**biinna (him and her)
- Pronouns that can be used to ask questions are classified as interrogative pronoun and tagged by PRONI. For example: **koni** (who), **amo** (what), **aabi** (which)
- Pronouns attached with preposition classified under pronoun preposition sub-class and tagged as PRONPREP. For example: no**och**(for us)
- All pronouns that cannot be classified under the above sub-classes are tagged with PRON.

Verb and its sub-classes

In this main class we identify one general class and four sub-classes of a verb. This class and its sub-classes are explained in the following example.

- Auxiliary verbs are one sub-class of verb and tagged as AUX without considering any attribute like gender, number, etc. For example: Mesfini paayileto **tunete** (Mesfen became a pilot)
- Verbs attached with conjunction categorized under verb conjunction sub-class and tagged by VC. For example: maao**hi**uchiye (ate and drank)
- Verbs that show infinitive tagged as VI. For example: dic**hoo**(to develop)
- Sometime an infinitive verb may be attached with conjunction. This type of verb classified under infinitive verb with conjunction sub-class and tagged by VIC. For example: xiishi**yoonaa** (to assure and)
- Verbs that cannot be classified under the above classification, they are categorized under the general tag i.e. verb and tagged by V.

Adverb and its sub-classes

Kafi-noonoo adverbs are words that qualify or modify a verb, adjective or other adverbs. In this thesis, we identify one general class and two subclasses. This class and its sub-class are explained using the following example.

- Adverbs attached with conjunction are tagged by ADVC. For example: shatiyoo**naa** (warning and)
- Adverbs attached with preposition are tagged as ADVPREP. For example: ame yawo**ona** (in what way)
- Other forms of adverbs that cannot be classified under the above classifications are tagged as ADV.

Adjective and its sub-classes

In this main class, we identify one general class and three sub-classes. This class and its sub-classes are explained in the following example.

- Adjectives attached with conjunction are tagged by ADJC. For example: maccoo**naa** ikkonoomee (social and economical)
- Adjectives attached with possession are classified under adjective possession sub-class and tagged as ADJPOSS. For example: de‟oyich (to take)
- Adjectives attached with preposition are tagged as ADJPREP. For example: digoo**yich** (for peaceful)
- All other forms of adjectives that cannot be classified under the above classification are tagged by ADJ.

Numerals

Kafi-noonoo numerals like that of English, Amharic and Afaan Oromo can be cardinal or ordinal which is tagged as CARDN and ORDN respectively. Numerals can be attached with conjunction and preposition. If the cardinal attached with conjunction and preposition, they are tagged with

CARDNC and CARDNPREP respectively. If the ordinal attached with conjunction and preposition, they are tagged with ORDNC and ORDNPREP respectively.

Preposition and its sub-classes

In Kafi-noonoo, prepositions alone does not convey any meaning unless they are attached with other basic classes like noun, adjective, adverb, etc. In this thesis; we try to identify preposition as a general class and preposition attached with conjunction as a subclass. This class and its sub-class are explained in the following example.

- Prepositions attached with conjunction are classified under preposition conjunction and tagged as PREPC. For example: **daggeexo** (at the middle) or **gasheexo** (at the edge)
- Prepositions other than the one mentioned above are tagged by PREP.

Conjunction

Conjunctions are words that serve to connect words, phrase, clauses or sentence. In this thesis, conjunctions that can be used as a separate word are tagged as C. Example: guutii (after doing something) woyee (or).

Interjections

Interjections are words used to express strong feeling or sad emotion. All words that show this type of characteristics are tagged as INT. Example: abo, wona.

Punctuations

All Kafi-noonoo punctuation marks such as.,?,!, „ and “ are tagged by PUNC.

The summarized version of Kafi-noonoo tagsets which are used to tag untagged Kafi-noonoo texts are shown in Table 3.5.

Table 3.5 Summary of Kafi-noonoo tagsets

No	Basic class	Derived class	Description	Example
1	Noun	N	Noun	Mixo
2		NP	Proper noun	Boonga
3		NPC	Proper noun + Conjunction	Yeerin
4		NPOSS	Noun + Possession	Ashich
5		NC	Noun + Conjunction	Mixooch
6		NPREP	Noun + Preposition	Ashicho
7	Pronoun	PROND	Demonstrative Pronoun	Ebi
8		PRONC	Pronoun + Conjunction	Biyaa
9		PRONI	Interrogative Pronoun	Amo
10		PRON	Pronoun	Taane
11		PRONPREP	Pronoun + Preposition	Nooch
12	Verb	AUX	Auxiliary Verb	Indiyoono
13		VC	Verb + Conjunction	Getaa
14		VI	Infinitive Verb	Dichoo
15		VIC	Infinitive Verb + Conjunction	Xiishiyoonaa
16		V	Verb	Wone
17	Adverb	ADV	Adverb	Kaawaa
18		ADVC	Adverb + Conjunction	Cokkeshoon
19		ADVPREP	Adverb + Preposition	Yawoona
20	Adjective	ADJ	Adjective	Cello
21		ADJPOSS	Adjective + Possession	De'oyich
22		ADJC	Adjective + Conjunction	gochiti
23		ADJPREP	Adjective + Preposition	Qaabbeechi
24	Numerals	CARDN	Cardinal	Ikkoo
25		ORDN	Ordinal	Guttinnee
26		CARDNPREP	Cardinal + Preposition	Ikkooch
27		CARDNC	Cardinal + Conjunction	Ikkoon
28		ORDNC	Ordinal + Conjunction	Ikkinoonaa
29		ORDNPREP	Ordinal + Preposition	Guttinneechi
30	Preposition	PREP	Preposition	Toomooch
31		PREPC	Preposition + Conjunction	Deshoon
32	Conjunction	C	Conjunction	Woyee
33	Interjection	INT	Interjection	Abo
34	Punctuation	PUNC	Punctuation	.,?,;

CHAPTER FOUR

Design of Kafi-noonoo Part-of-speech Tagger

In this Section, a detailed description of design issue and techniques of Kafi-noonoo part-of-speech tagger development is presented.

4.1 System Architecture

Several approaches have been proposed to annotate words automatically with their part-of-speech tags. Among these, the hybrid of HMM and rule-based approach is assumed to perform better than the HMM and rule-based taggers taken alone. For this thesis, a hybrid approach, which is a combination of HMM and rule-based tagger at sentence level is designed for Kafi-noonoo language.

The hybrid tagger of Kafi-noonoo consist of three main components these are initial state (HMM tagger), output analyzer and rule-based tagger. The overall architecture of the system including the connection between the components are shown in Figure 4.1.

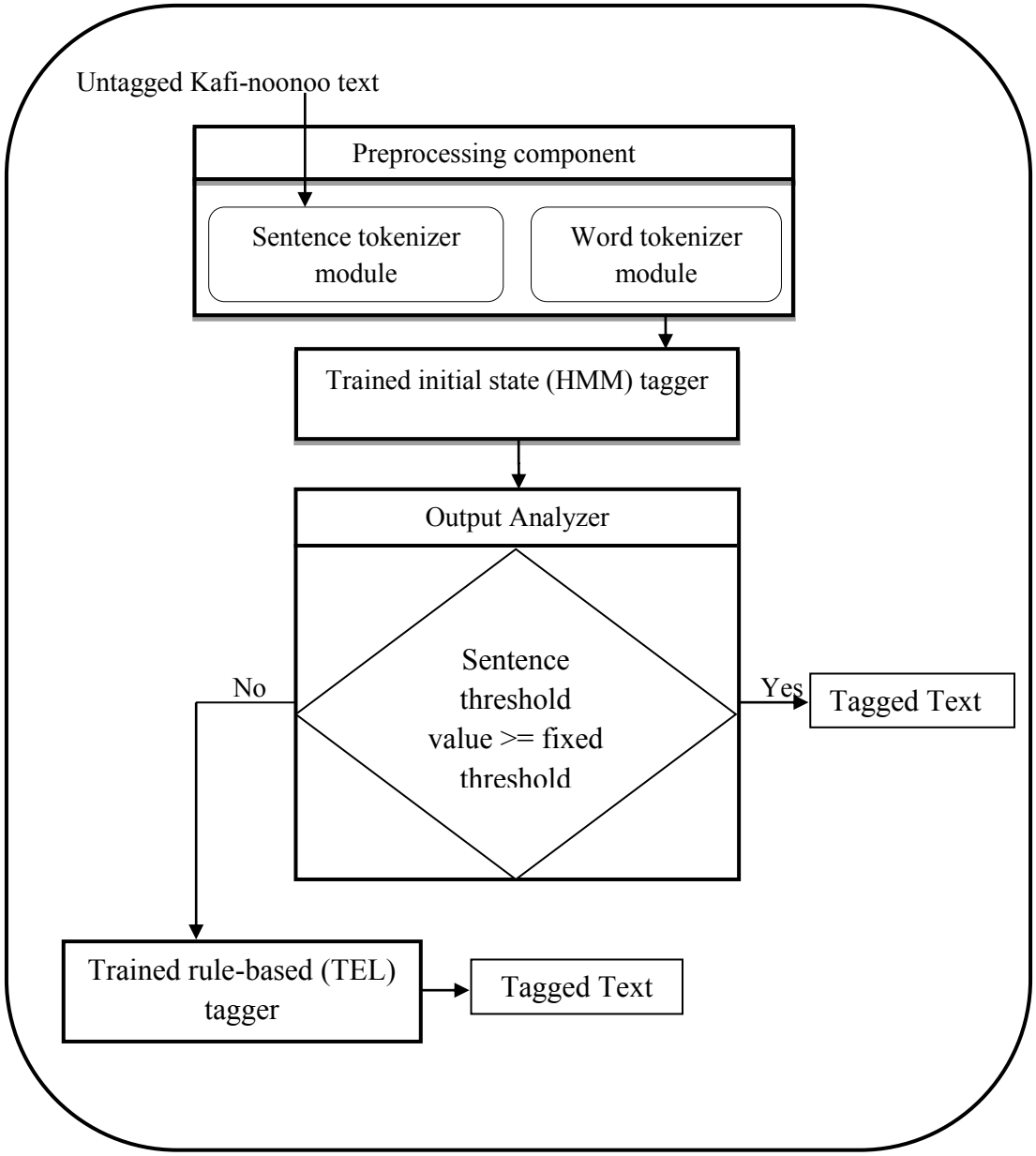

Figure 4.1 Architecture and components interconnection of the hybrid system

Figure 4.2 shows the algorithm of the hybrid tagger.

1. *Read the text to be tagged(**input**)*
2. *Set experimental threshold value*
3. *Get trained HMM tagger*
4. *Get trained TEL tagger*
5. *Tag the untagged text with HMM tagger*
6. *Get the probability of each tag given the word while HMM tagger is assigning the tag for the word*
7. *While there are HMM tagged sentence*
   - *7.1. Add the probability of each word within the sentence and divide with the number of token within the sentence*
     - *7.1.1. If the probability of the entire sentence > = fixed threshold value then*
       - *7.1.1.1. The sentence tagged with the trained HMM tagger considered to be the correct one (**output**)*
     - *7.1.2. Else*
       - *7.1.2.1. Apply TEL tagger to the entire sentence*
       - *7.1.2.2. The sentence that is tagged by Trained TEL tagger considered to be the correct one (**output**)*
8. *End of HMM tagged sentence*

Figure 4.2 Algorithm for the hybrid tagger of Kafi-noonoo language

4.1.1 Statistical Component of the Tagger

In order to develop the statistical part of the hybrid system or tagger, we adopted an HMM method. It is the most common method under statistical approach. It find the optimal part-of-speech tag sequence for a given word sequence, using a Viterbi algorithm [5].

The Viterbi algorithm is a dynamic programming algorithm that applies a table driven method to solve problems by combining solutions to sub problems [5]. It reduce the complexity of HMM tagger in terms of time and memory consumption by keeping only the best sub-path of each node at each position in the sequence and discarding the others.

To find the best tag sequence for an entire sentence, the method uses the following formula [5].

$$\hat{\mathrm{T}} = \underset{T}{argmax}\, P(T/W) \tag{4.1}$$

Where

$\hat{T}$ is sequence of tag

T is the most probable tag sequence

W is the sequence of word in the sentence

$\underset{T}{argmax}\, P(T/W)$ is a function given by T such that P(T/W) is largest

According to Bayes‟ law [5], P(T/W) can be expressed as:

$$\mathrm{P(T/W)} = \frac{\mathrm{P(T)P(W/T)}}{\mathrm{P(W)}} \tag{4.2}$$

Based on this, Equation 4.1 can be rewritten as:

$$\hat{\mathrm{T}} = \underset{T}{argmax}\, \frac{P(T)P(W/T)}{P(W)} \tag{4.3}$$

Since we are looking for the most likely tag sequence of a sentence given a word sequence, the probability of the entire sentence i.e. P(W) has a constant value for each tag sequence within the sentence and we can ignore it. As a result Equation 4.3 can be rewritten as:

$$\hat{\mathrm{T}} = \underset{T}{argmax}\, P(T)P(W/T) \tag{4.4}$$

According to the chain rule of probability [5], P(T)P(W/T) can be computed as:

$$P(T)P(W/T) = \prod_{i=1}^{n} P(w_i/w_1 t_1 \dots w_{i-1} t_{i-1} t_i)\, P(t_i/w_1\, t_1) \dots w_{i-1} t_{i-1} \quad (4.5)$$

Where w_i is the i^{th} word in the given sentence and t_i is the i^{th} tag in the tagset.

Due to too many possible sentences, it is difficult to compute the chain rule probabilities through count and divide mechanism. In order to compute these probabilities, we make the following assumptions.

1. The probability of a word is dependent only on its tag (Markov simplifying assumption).
2. Tag history can be approximated by the most recent tag

Based on these assumptions, Equation 4.4 can be rewritten as:

$$\hat{T} = \underset{T}{argmax} \prod_{i=1}^{n} P(t_i/t_{i-1})\, P(w_i/t_i) \quad (4.6)$$

The first factor of this product is called lexical model and the second factor is called the contextual model.

The lexical model computes lexical probabilities of each word in the training data. In order to calculate these probabilities, we use maximum likelihood estimation from relative frequencies using the following formula [16].

$$P(w_i/t_i) = \frac{C(w_i,t_i)}{C(t_i)} \quad (4.7)$$

The contextual model which is also called the N-gram model computes contextual probabilities (information about the context where the word is found in the given sentence). These can be achieved by maximum likelihood estimation from relative frequencies as [5]:

$$P(t_i/t_{i-1}) = \frac{C(t_{i-1},t_i)}{C(t_{i-1})} \quad (4.8)$$

By changing the value of n, we can generate different N-gram model. Most of the time, in POS tagging development, n is equal to 2 or 3. The choice of value for n is dependent on the training data. According to [13], bi-gram model is preferable for large tagset or small training data. Based

on this, for this thesis, we choose n to be equal to 2. This yields a contextual model that considers only the previous one tag.

To train the statistical part of the system, we use a supervised learning mechanism, where a pre-tagged corpus is a prerequisite. The process follows three steps. In the first step, we provide a tagged corpus as a training data. The tagged data passed through a pre-processing module (sentence and word tokenizer) to be tokenized both at sentence and word level respectively. In the second step, the output of the pre-processing component passed to the statistical analyzer module that computes both lexical and contextual probabilities and stores them in the database. Finally, we provide list of tags and store them in the database. The overall training process of the statistical component of the system is shown in Figure 4.3.

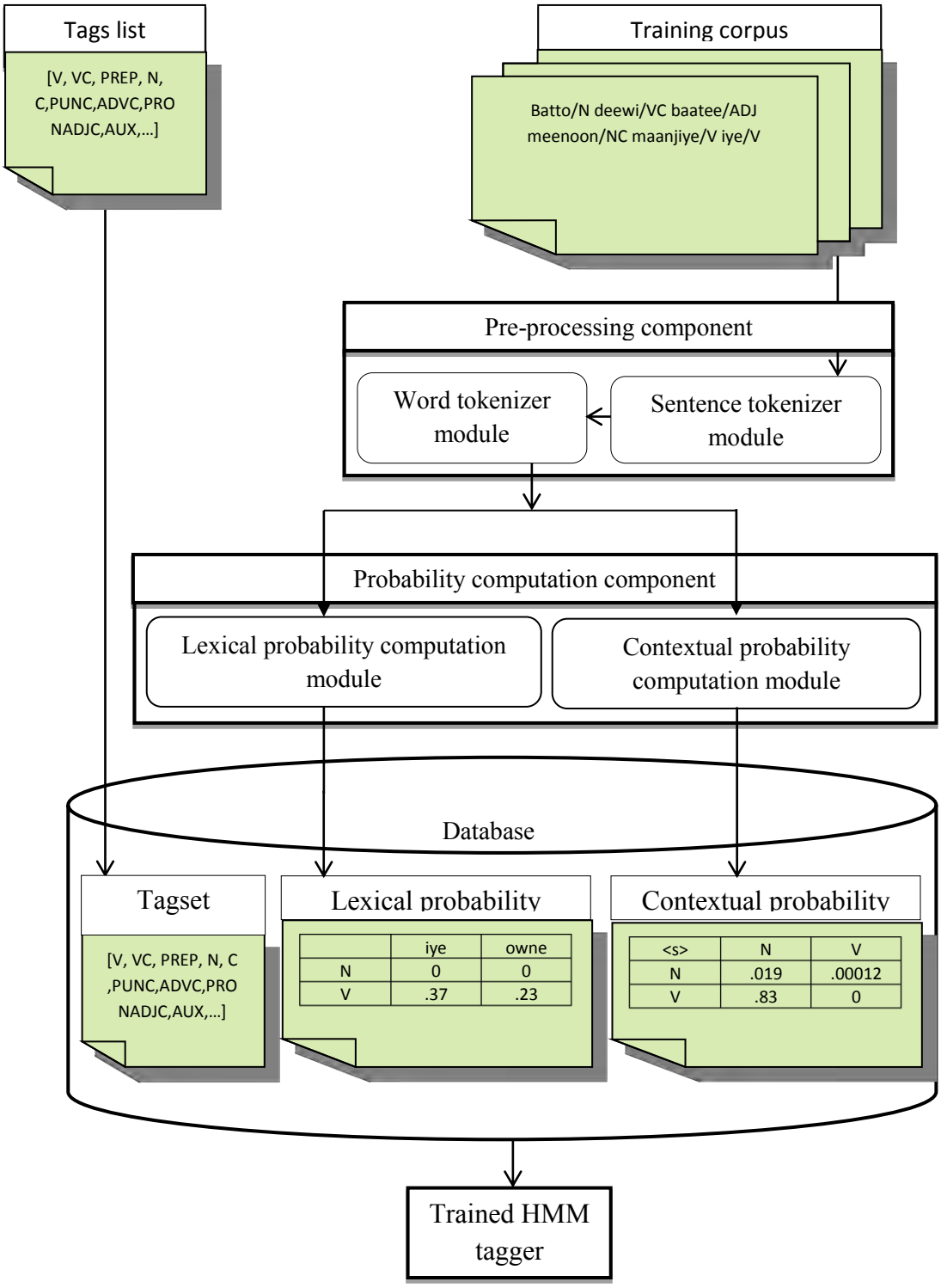

Figure 4.3 Training process for statistical component of the system

After training the statistical component of the tagger, it is used as an initial-state tagger for the entire system.

The lexical and contextual probabilities that are stored within the database provide information about the probability of each tag given the word and information about the context of the word respectively. Based on the stored information, the initial-state tagger assign an optimal part-of-speech tag for a given word sequence using a table driven method called Viterbi algorithm and provide the tagged text as an output. The pictorial representation of initial-state tagger (HMM tagger) tagging process is shown in Figure 4.4.

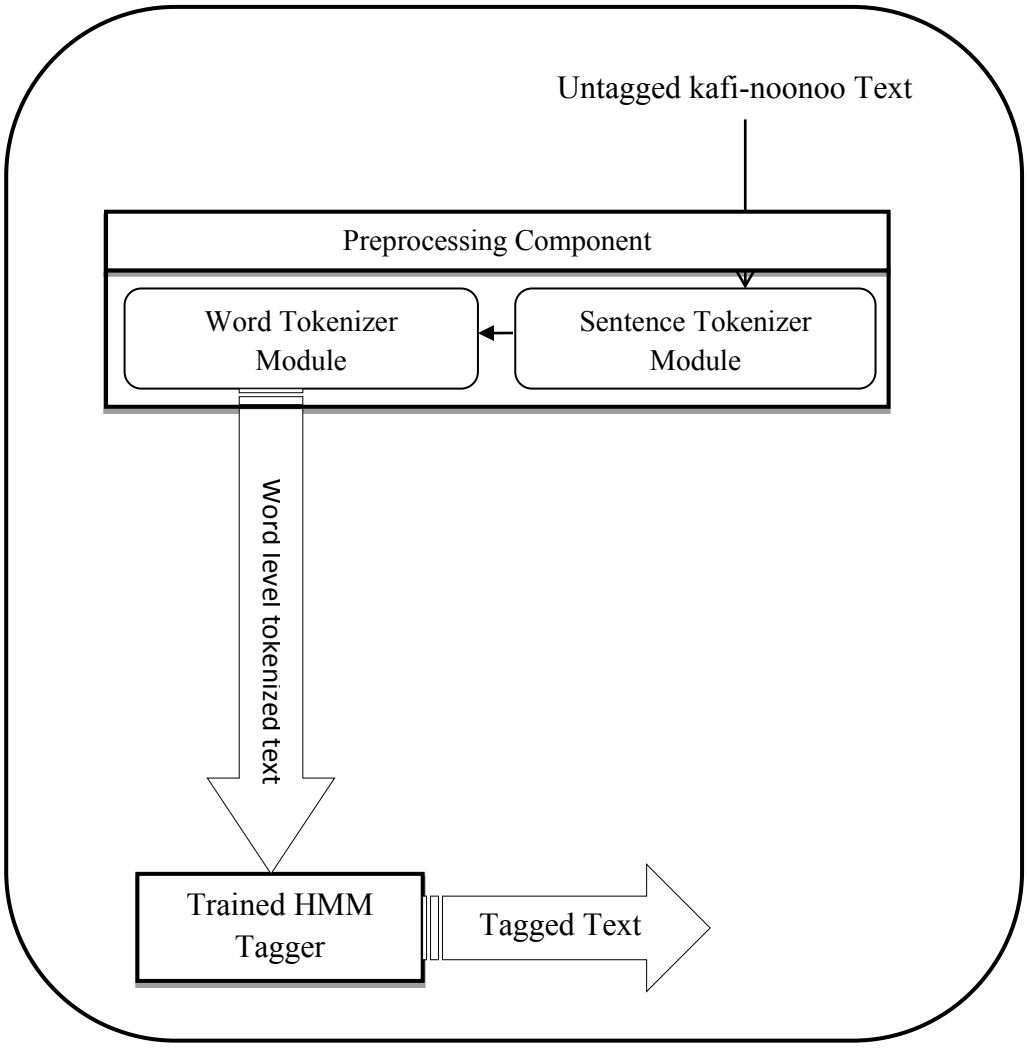

Figure 4.4 Initial-stage tagger tagging process

The lexical probability that is obtained at training time assigns a probabilistic value for each word within the sentence. This enables us to compute the probability of the entire sentence using the other main component of the system called output analyzer.

4.1.2 The Output Analyzer Component

It is a major component within the system that performs two main operations: sentence level probability computation (confidence level of the sentence) operation and value comparison operation. The first operation is performed by adding the probability of each tag given the word (lexical probability) of each word within the sentence and divide by the length of the sentence. This can be expressed mathematically as:

$$\frac{\sum_{i=0}^{len(sent)} P(W_i/T_i)}{len(sent)} \tag{4.9}$$

Where

$P(W_i/T_i)$ is the probability of tag given a word

len(sent) is the length of the sentence

In the second main operation, the output analyzer decide whether or not the confidence level of the entire sentence that is obtained from the first main operation of the output analyzer is greater than or equal to that of the fixed threshold value. The threshold value is a predetermined value used for checking the confidence level of tagging a given sentence. If the threshold value of the entire sentence greater than or equal to that of the predetermined threshold value, the sequence of tags that are assigned for the entire sentence does not need any further correction. Otherwise, the entire sentence passed to the next component of the system called rule-based component.

4.1.3 Rule-Based Component of the Tagger

It is a major component of the system that performs the same functionality like that of the statistical component of the system with different approaches. In order to develop this component, a TEL approach is adopted with a little bit modification within the learners‟ templates to fit with Kafi-noonoo language features such as prefix, the maximum length of character that can be deleted from the begging of the words to predict tag of unknown words

from the existing one and the length of word/tag that can be allowed before and/or after the given token to find the tag of the token based on contextual information etc. Figure 4.5 shows the overall structure of the adopted transformation-based error-driven learning approach for Kafi-noonoo language.

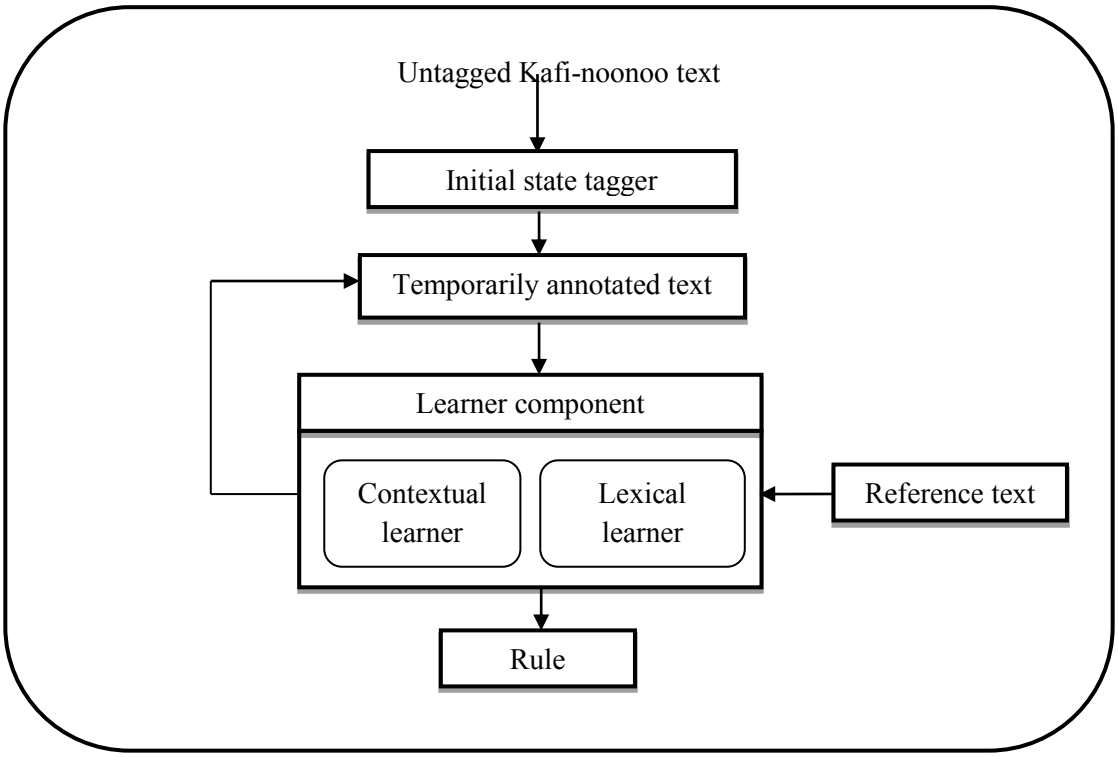

Figure 4.5 Adopted TEL approach for Kafi-noonoo language [7]

It has three major sub-components: initial state tagger, learner and rule sub-components. The detailed description of each sub-component is given below.

Initial-State Tagger

The initial state tagger component takes untagged Kafi-noonoo text and tagged with their most likely tag. In case of part-of-speech tagging, different initial-state tagger can be employed that ranges from stochastic n-gram tagger that assigns the most likely tag as indicated in the training corpus to default tagger that label all words as nouns.

Learner

The learning component of the adopted TEL approach has two sub-components: lexical and contextual rule learner sub-components.

Lexical Rule Learner

The lexical rule learner is used to drive lexicon and rule that assign the most likely tag for a given word that may or may not be seen in the training corpus.

The lexicon is computed using a statistical method and it contains every word within the training corpus associated with its most frequent tag. It is used to tag untagged words that are seen at list one time during the training phase.

In order to generate the lexical rules, the lexical rule learner takes untagged Kafi-noonoo text and passes it through the initial-stage tagger to produce a temporary corpus called KTC_0. Following this, based on the condition which is predefined in the lexical rule learner template, it finds the rule which gets the best permissible score when applied to KTC_0. A best score for a rule means that a rule that gives better resemblance with the reference text when applied to KTC_0. It can be computed as follows: for each tagged words in the temporary corpus (KTC_0), the rule gets a score for that word by comparing the change from current tag to the resulting tag with respect to the word within the reference text. Based on the effect of the rule on the word to be tagged, the score of the rule may become positive, negative or zero.

- Positive (+): The rule change the tag of the word from incorrect to correct
- Negative (-): The rule change the tag of the word from correct to incorrect
- Zero (0): Condition of the rule not satisfied

After computing rule with best score, it is applied to the first temporary corpus (KTC_0) in order to produce the next temporary corpus called (KTC_1) and added to the set of rules. The process continue in the same fashion to produce all the permissible rule with the corresponding temporary text until no rule can further improve the tag of the temporary corpus.

Templates that are used in lexical rule learner component are given below

1. Change the most likely tag to Y if the current word has suffix X
2. Change the most likely tag to Y if deleting/adding the suffix X, |X|<3, results in a word, |X| is length of x.
3. Change the most likely tag from X to Y if deleting the prefix (character) X, |X|<3, results in a word, |X| is length of x.
4. Change the most likely tag from X to Y if word W ever appears immediately to the left/right of the word.
5. Change the most likely tag to Y if the character Z appears anywhere in the word.

Contextual Rule Learner

Once the lexical rule learner sub-component of the rule-based tagger builds and learned a lexicon to tag each word similar to the one found in the tagged training set and a lexical rule for predicting the most likely tag for unknown words, contextual rules are learned for disambiguation and better accuracy using contextual rule learner sub-component. In order to make accurate prediction of tags for words, the contextual rule learner finds rule on the basis of context of the word. In order to generate contextual rule, first the learner accepts both temporary and reference text as an input. Then, the learner generates all possible rules from the predefined contextual rule template when the trigger is satisfied. A trigger is a set of predefined conditions in the form of templates that must be satisfied (came true) to generate the contextual rule.

Triggers that are used in the contextual rule learner template are given below.

1. The preceding/following word is tagged with X
2. One of the two preceding/following words is tagged with X
3. One of the three preceding/following word is tagged with X
4. The preceding word is tagged with X and the following word is tagged with Y
5. The preceding/following two words are tagged with X and Y
6. The two words before/after is tagged with X

After generating all possible rules, the learner computes the score of each rule for a particular word. Based on the score, the learner picks rules with highest score and stores it in the sub-component of rule-based tagger called rules.

For each word W in the temporary corpus, the learner computes the score. These can be achieved by comparing the tag of words in the temporary corpus after applying the rule with that of the reference text. If the rule is applied to the word and corrects an error, the score of the rule is +1, while the rule introduces an error, the score of the rule is -1, and otherwise the score of the rule is 0. The total score of each rule are computed by adding score of the rule when it is applied to each word within the temporary corpus.

Following this, the learner takes the rules that are stored in the rule component of the tagger and apply them on the temporary corpus to generate another temporary corpus. The process continues in the same fashion until no rule exists that makes the temporary corpus resemble with that of the reference text.

Rule

It is the sub-component of TEL (rule-based) tagger with two main parts namely triggers (condition or current tag) and rewrite (resulting tag). Figure 4.6 shows the main parts of a rule.

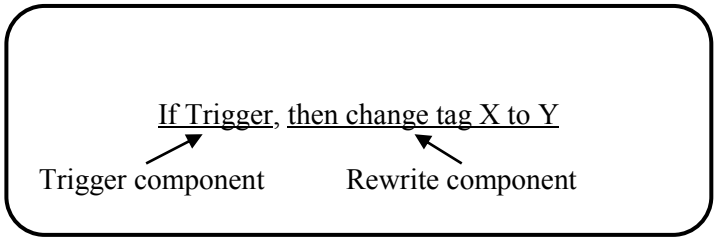

Figure 4.6 Main parts of a rule

Once the adopted TEL approach trained on the training corpus and learns set of lexical and contextual rules using lexical and contextual rule learner sub-components of the rule-based tagger, the rule sub-component of the rule-based tagger store each learned rule within a file. In addition to these, the rule-based component builds a lexicon to tag words that are seen at least once during training time. All rules and dictionaries are stored in a file. As a result, the rule-based component of the system becomes a trained model that is able to tag the untagged texts of Kafi-noonoo language using the stored information at the time of training. The pictorial representation of the rule-based component (TEL tagger) of Kafi-noonoo POS tagger is shown in Figure 4.7.

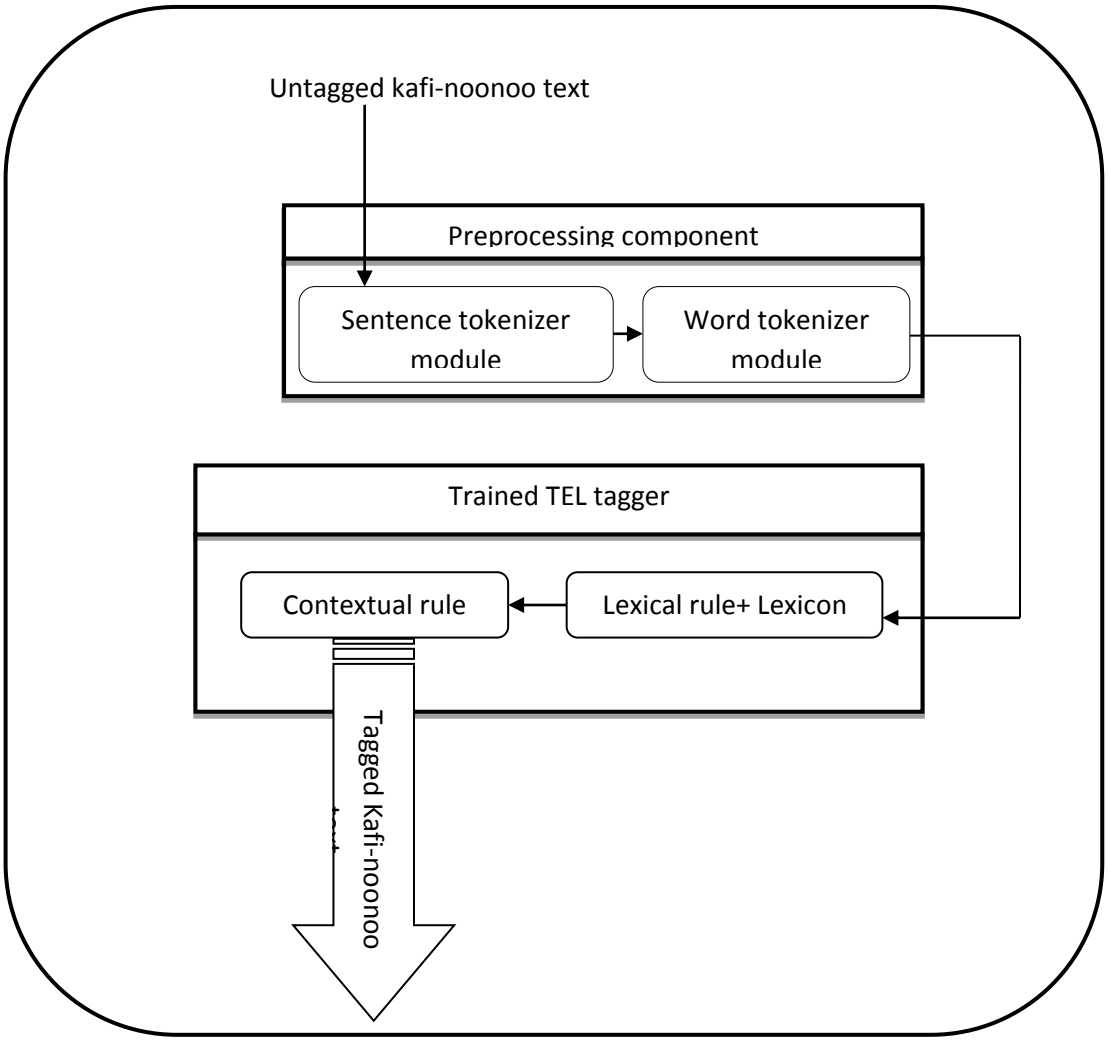

Figure 4.7 Rule-based tagger tagging process for Kafi-noonoo language

In general, the hybrid tagger works in three main steps. In the first step, the statistical component (HMM tagger) of the system tag the given raw texts and provide probability of each tag given the word. Based on each word probability, the output analyzer component computes the threshold value of the entire sentence. If the threshold value of the sentence is greater than or equal to the fixed threshold value, the assigned sequence of tags within the sentence does not need any correction. Otherwise, the entire sentence is passed to the rule-based component (TEL tagger) for correction.

CHAPTER FIVE

Experiment

5.1 Introduction

Kafi-noonoo part-of-speech tagger is implemented for corpus from two different genres using Natural Language Toolkit and Python. The reason behind the choice of these tools is their suitability for processing different NLP tasks [3].

NLTK is an open source toolkit that contains open source python modules, linguistic data and documentation for research and development in natural language processing field [27]. It supports many NLP tasks such as POS tagging, classification, chunking and parsing for all major operating systems: Windows, Linux/Unix and Mac. It also contain different corpus for different languages such as English, German and French.

Python is a simple but powerful programming language with excellent functionality for processing linguistic data [3, 24 and 27]. It has efficient and high level data structure with simple but effective approach to object-oriented programming [27]. Moreover, its syntax and dynamic typing feature with its interpreted nature makes it a powerful language for scripting and rapid application development [27].

This chapter presents details about corpus preparation, implementation of the pre-processing components and experimental results of HMM, rule-based and hybrid tagger of Kafi-noonoo language.

5.2 Corpus Preparation

Corpus is a collection of large text with or without additional linguistic information. Corpus with additional linguistic information is known as tagged (annotated) corpus [5, 22]. The tagged corpus can be used as inputs to many NLP applications such as part-of speech tagging, sentimental analysis and parsing.

Based on the genre included within the corpus, corpus classified in to two balanced and category specific corpus. Developing a balanced corpus increases the performance of the tagger by providing many words taken from different category. But it need time, money and skill of

language expert. In case of category specific corpus, the tagger trained on a text taken from only one category and if a text taken from different category provided to the trained tagger, the performance of the tagger may not be as expected. Due to the above mentioned constraints, the corpus developed for this thesis contains two genre: lore and academic.

As far as the researcher knowledge is concerned, let alone a balanced corpus, there is no category specific corpus developed for Kafi-noonoo language. To develop the tagged version of Kafi-noonoo corpus, a flat text is tagged with the corresponding part-of-speech tags using an incremental corpus preparation approach.

Incremental corpus preparation (semi-automatic) approach involves three main stages, manual, automatic and correction stage. In the manual stage, the text collected from different source passed to the annotators for manual annotation and the output of the manual annotation is used to train the tagger (HMM/Brill tagger). After this, in the automatic stage, the tagger, that is trained on the manually annotated text, tag new raw text. In the correction stage, the output of the trained tagger is passed to the annotator for manual correction. The output that is obtained from the correction stage is added to the training set to contain the approved text which is used for training the tagger. Starting from the automatic stage, the process is repeated until the desired corpus size is achieved. Figure 5.1 shows the stages involved in the incremental corpus preparation approach for word class tagged corpus preparation.

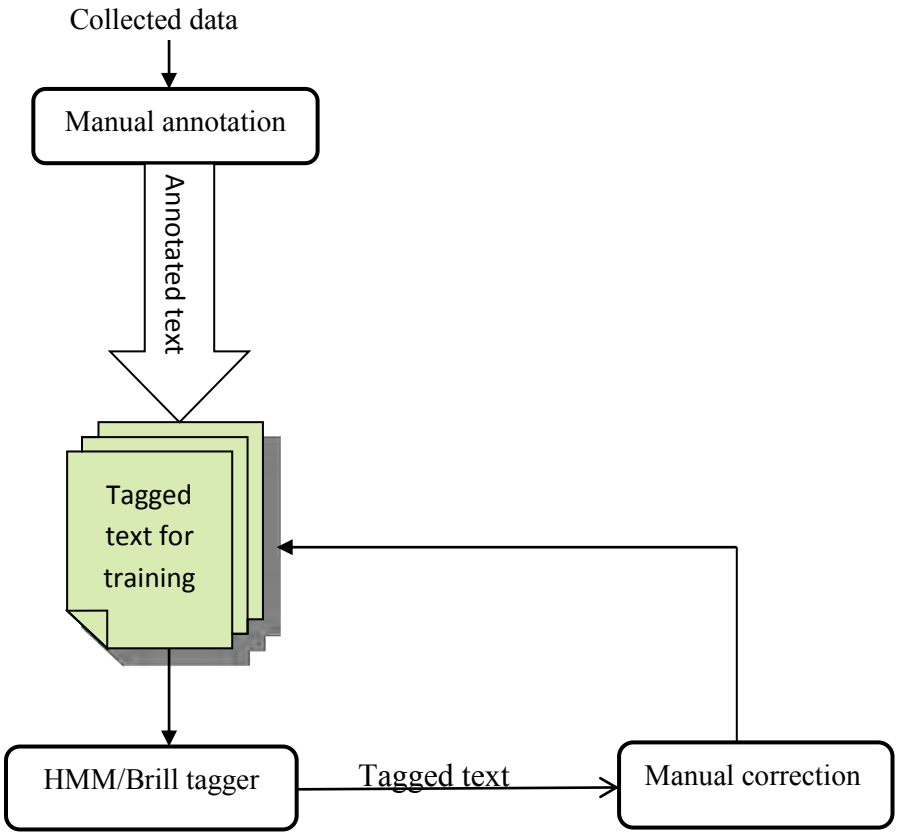

Figure 5.1Steps involved in the incremental corpus preparation process

Among the three stages within the incremental corpus preparation approach, the manual stage is tedious and time consuming. Therefore, to start the corpus preparation process, only one hundred flat Kafi-noonoo sentences are given to be tagged manually by the annotator. A sample Kafi-noonoo corpus with flat and tagged version is shown in Appendix A.

5.3 Pre-processing Component

As we have seen in Chapter Four, Section 4.1, 4.1.1, and 4.1.3, the sub-components (pre-processing, HMM and rule-based components)that builds the hybrid tagger have two main modules, sentence and word tokenizer module. The sentence splitter module accepts both tagged and untagged texts using Kafi-noonoo corpus reader and splits down at sentence level based on Kafi-noonoo sentence end marker characters. Afterwards, the word tokenizer module tokenizes the output of sentence splitter module into word level. In the training phase, the tokenized word comprises two components, word/token and part-of-speech tag. The word and its part-of-speech

tag is separated using forward slash (/) character. This enables the tagger to compute statistical information for both word and part-of-speech tag during the training phase.

5.4 Test Results

Several experiments with different training set on different part-of-speech tagger have been conducted for Kafi-noonoo POS tagger. To do these, the entire corpus is divided into two main sets: training set and testing set. The training set covers 90% of the entire corpus. The remaining 10% of the corpus is used for testing purpose.

5.4.1 Test Result of HMM Tagger

To see the goodness of the training set, we conducted ten different experiments with different portion of the training set using NLTK based HMM tagger. Originally NLTK corpus reader works on corpora that are tagged using angle brackets (<>). To read the corpora that are tagged with their respective POS tags using forward slash (/) character, we made a little bit modification on the corpus reader. In order to conduct the experiments, first the researchers divide the entire training set in to ten parts. Then the researchers train the system using 10% of the training set. After training, the performance of the trained system is measured using the test corpus. This process is repeated by adding the training data by 10% until they got a desired performance of the tagger. In fact, the desired performance of the tagger is considered to be the performance measured from the learning curve when all the training set (100%) is utilized. Table 5.1 shows the result of different experiments conducted on different portion of the training set with their corresponding performance.

Table 5.1 HMM tagger performance

Training set	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Performance (%)	57.32	62.54	63.51	68.4	69.7	70.68	71.98	72.31	75.5	77.19
Difference	57.32	5.22	0.97	4.89	1.3	0.98	1.3	0.33	3.19	1.69

Figure 5.2 shows the performance curve of HMM tagger.

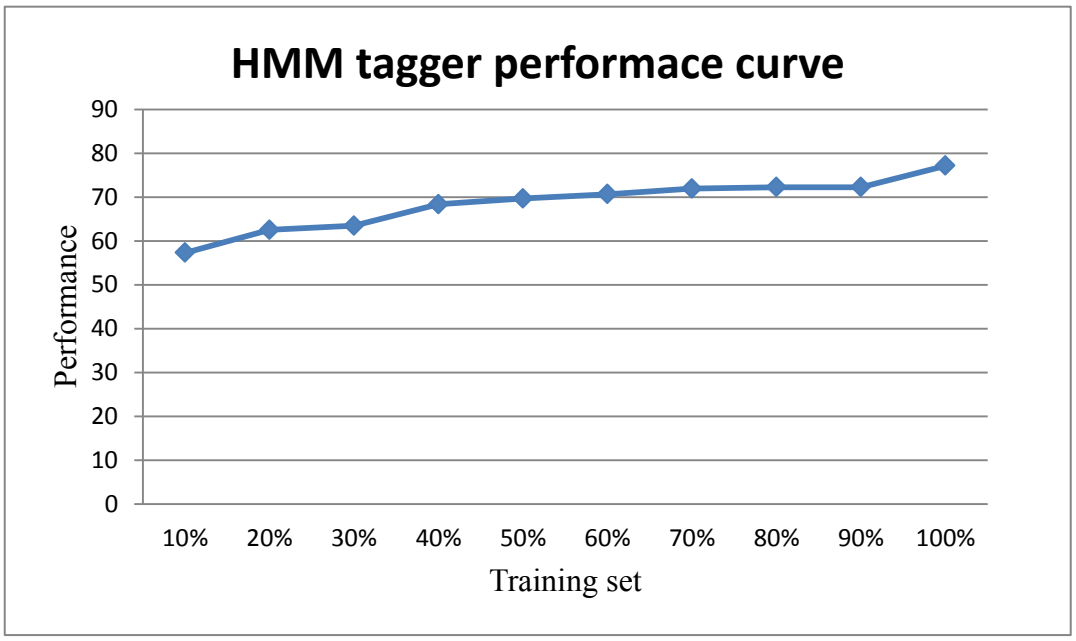

Figure 5.2 Performance curve analysis for HMM tagger

5.4.2 Test Result of Rule-based Tagger

To test the performance of the rule-based tagger like that of HMM tagger, ten different experiments are conducted using different portions of the training set with two different initial-state tagger namely default and unigram tagger. When the rule-based tagger uses default tagger as initial state tagger, it assigns their tag to every single word, even for words that have never been encountered before. For this thesis, based on frequency of individual tag within the training set and nature of the Kafi-noonoo language, the researchers identify verb (V) as a default word class for the initial-state tagger. While in case of unigram tagger, it assigns the most likely tag for a given word based on the frequency of individual tag within the training set. Table 5.2 shows the different experiments conducted using different portions of the training set with the corresponding performance of the rule-based tagger for different initial state taggers: default, and unigram taggers.

Table 5.2 Performance of rule-based tagger with different initial state tagger

Training set	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Default tagger	56.35	56.35	56.67	56.02	57.32	59.28	60.62	60.26	59.60	61.23
Unigram tagger	40.36	41.36	43.97	46.25	49.83	52.44	52.76	55.37	55.37	61.88

Figure 5.3 shows the performance curve of rule-based tagger

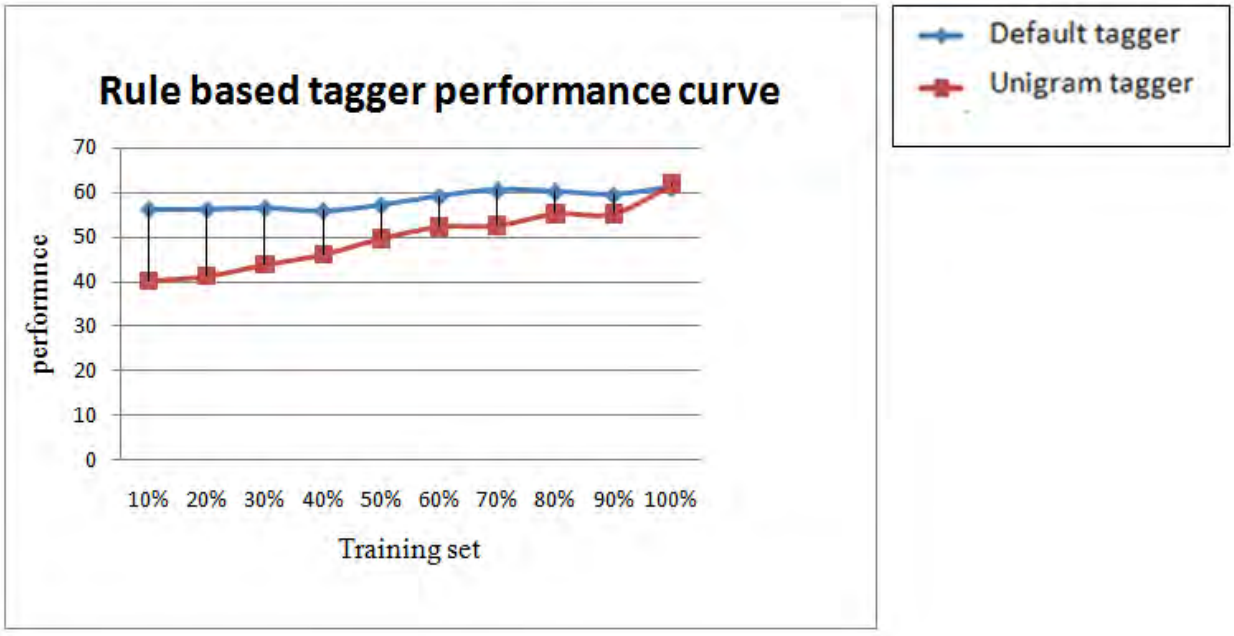

Figure 5.3 Rule-based tagger performance curve analysis

5.4.3 Test Result of Hybrid Tagger

Hybrid tagger of Kafi-noonoo language is composed from two different taggers, HMM and rule-based tagger. In order to tag a given text with the hybrid tagger, first the HMM tagger annotates the word sequence within the sentence and if the desired threshold value of the entire sentence is not achieved, the entire sentence is passed to the rule-based tagger for correction. After conducting different experiment with different threshold value, the threshold value is fixed to 0.56 since taking threshold value less than 0.56 does not bring significant difference on the performance of the tagger. As we can see from Table 5.3, when the threshold value increase, the rule-based tagger corrects more words; as a result, the hybrid tagger scores highest performance when the threshold value goes up. By setting the threshold value to 0.56, an overall performance

of 80.47% is obtained. Table 5.3 show the performance of hybrid tagger with different threshold value.

Table 5.3 Performance of hybrid tagger with different threshold value

Threshold value	0.02	0.36	0.48	0.56	1
Performance (%)	77.19	80.47	80.47	80.47	79.31

Figure 5.4 shows the performance analysis of hybrid tagger with different threshold value.

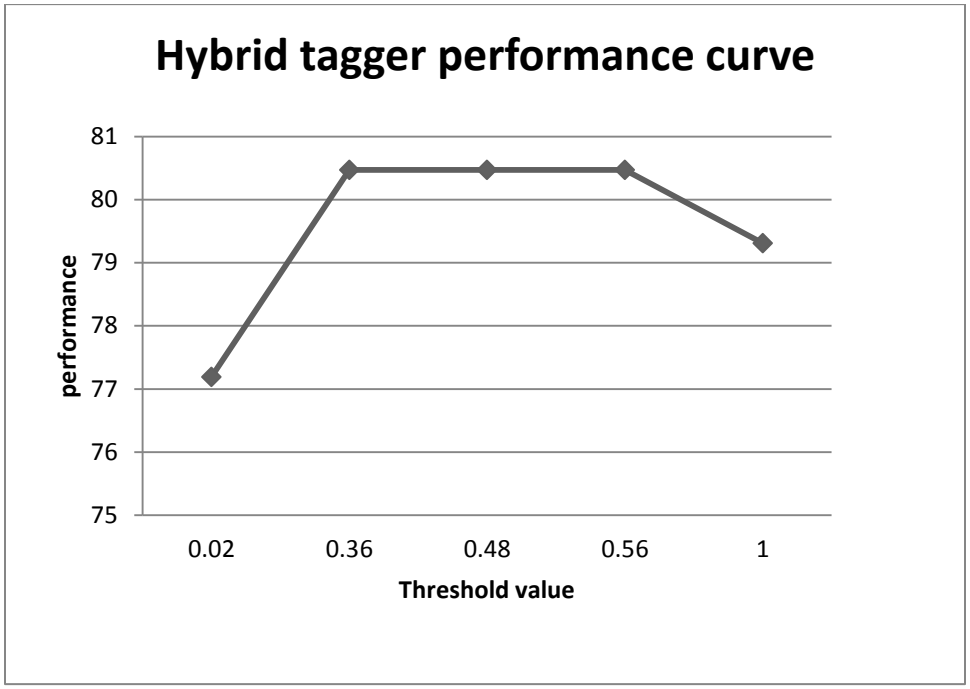

Figure 5.4 Performance analysis of hybrid tagger with different threshold value

5.5 Performance Analysis

To analyze the performance of HMM, rule-based and hybrid tagger of Kafi-noonoo language in relation with different part-of-speech tags, a frequency of tags within the total corpus, training set and testing set are considered. In addition to this, a confusion matrix is developed for HMM, rule-based and hybrid taggers. Based on these, the entire tags are divided in to two parts, the 9 most frequent tags and the reset as OTHERS. Table 5.4 shows the frequency of tags in relation with total corpus, training set and testing set.

Table 5.4 Frequency of tags

Tag	Tag frequency within total corpus	Tag frequency with in training set	Testing set	
			Tag frequency	%
V	1084	976	108	9.96
VC	478	438	40	8.36
N	461	412	49	10.63
PUNC	375	335	40	10.67
NC	309	280	29	9.38
PRON	160	149	11	6.88
ADJ	132	122	10	7.58
PRONC	43	37	6	13.96
PREP	37	34	3	8.10
OTHERS	242	231	11	4.54
Total	3296	3015	307	9.31

Table 5.5 shows the confusion matrix of HMM based tagger for Kafi-noonoo language.

Table 5.5 Confusion matrix for HMM based tagger

		Test										Total	Performance (%)
		V	VC	N	PUNC	NC	PRON	ADJ	PRONC	PREP	Others		
Reference	V	**106**		1				1				108	98.14
	VC	6	**28**	4							2	40	70.00
	N	9	2	**36**				2				49	73.47
	PUNC				**40**							40	100.00
	NC	5	2	2		**20**						29	68.97
	PRON	3					**7**	1				11	54.54
	ADJ	2						**8**				10	80.0
	PRONC								**6**			6	100.00
	PREP						1			**2**		3	66.66
	OTHERS	3	1	1							**6**	11	54.55
	Total	134	33	44	40	20	8	12	6	2	8	307	77.19

The HMM based tagger confusion matrix shows that it assigns 259 tags correctly and 48 tags incorrectly to the tokens within the testing set. Due to lack of balanced and large corpus for training the system, it confused the tags to other part-of-speech tags; for instance out of 49 nouns

tokens, 13 are assigned incorrectly to other part-of-speech tag. The performance of the HMM based tagger varies from one part-of-speech tag to another part-of-speech tag. As we can see from Table 5.5,it performs better on tag PUNC and PRONC followed by tag V, ADJ, N, VC, NC, PREP, OTHERS, and PRON.

Table 5.6 shows the confusion matrix of rule-based tagger for Kafi-noonoo language.

Table 5.6 Rule-based tagger confusion matrix using unigram tagger as initial-stage tagger

		Test										Total	Performance (%)
		V	VC	N	PUNC	NC	PRON	ADJ	PRONC	PREP	Others		
Reference	V	**77**									31	108	71.29
	VC		**17**								23	40	42.5
	N			**26**							23	49	53.06
	PUNC				**40**							40	100.00
	NC					**9**					20	29	31.03
	PRON	1					**6**				4	11	54.54
	ADJ							**2**			8	10	20
	PRONC								**6**			6	100.00
	PREP						1			**2**		3	66.67
	OTHERS		2	2				1	1		**5**	11	45.45
	Total	78	19	28	40	9	7	3	7	2	114	307	61.88

The rule-based tagger confusion matrix shows that it assigns 190 tags correctly and 117 tags incorrectly. The performance of the rule-based tagger varies for the different part-of-speech tags with a higher performance for PUNC and PRONC part-of-speech tags followed by V, PREP, PRON, N, OTHERS, VC, NC, and ADJ for the given testing set.

Table 5.7 shows the confusion matrix of hybrid tagger for Kafi-noonoo language.

Table 5.7 Confusion matrix for hybrid tagger

		Test										Total	Performance (%)
		V	VC	N	PUNC	NC	PRON	ADJ	PRONC	PREP	Others		
Reference	V	**107**						1				108	99.07
	VC	4	**32**	3							1	40	80.00
	N	9	2	**36**				2				49	73.47
	PUNC				**40**							40	100.00
	NC	4	2	1		**22**						29	75.86
	PRON	3					**7**	1				11	63.63
	ADJ	2						**8**				10	80.00
	PRONC								**6**			6	100.00
	PREP						1			**2**		3	66.66
	OTHERS	3	1	1							**6**	11	54.55
	Total	132	37	41	40	22	8	12	6	2	7	307	80.47

The hybrid tagger confusion matrix shows that it assigns 266 tags correctly and 41 tags incorrectly. Due to language features, lack of standard corpus and incorrect labeling of token in the prepared corpus, it confused the tags to other part-of-speech tags. But as we can see from Table 5.7, the confusion made by the hybrid tagger is less than the confusion made by the individual tagger. Like HMM and rule-based tagger, the performance of the hybrid tagger varies from one part-of-speech tag to another part-of-speech tag. As we can see from Table 5.7 it performs better on tag V followed by tag VC, NC, and PRON than that of HMM and rule-based tagger.

CHATPTER SIX

Conclusion and Recommendation

6.1 Conclusion

Part of speech tagging is the process of classifying words into their part-of-speech and labeling them accordingly. It is a research area in the field of natural language processing for different languages. Several approaches have been proposed to annotate words automatically with their part-of-speech tags. Among these, the hybrid of HMM and rule-based approach is assumed to perform better than the HMM and rule-based taggers taken alone.

For this thesis, a hybrid approach, which is a combination of HMM and rule-based tagger at sentence level is designed for Kafi-noonoo language.

Corpora are important for many types of linguistic research including part-of-speech tagging. For this thesis, a corpus with a total of 354 sentences is collected from two genres. There are several standard tagset used in corpora and in POS tagging experiments. For this thesis, 34 part-of-speech tags are identified as a tagset for annotating a raw text. The tagset indicates only word class rather than gender, number, tenses etc.

The tagged corpus is divided into training and testing set. The training set comprises 90% of the entire corpus the reaming corpus is used as a testing set.

NLTK version 2.0.4 and Python version 2.6.6 are used in the implementation and experiment of Kafi-noonoo part-of-speech tagger. To test the performance, different experiments are conducted for the three types of taggers namely the HMM, rule-based and hybrid tagger. As a result, 77.19%, 61.88% and 80.47% performances are obtained for HMM, rule-based with unigram initial state tagger and hybrid taggers respectively. Therefore, it is possible to conclude that the hybrid tagger at sentence level performs better than HMM and rule-based tagger taken independently.

6.2 Recommendation

There are lots of research areas in natural language processing that can be done for local languages. Among these, part-of-speech tagging is a useful form of linguistic analysis. It serves as pre-processing component for many higher levels NLP applications such as spelling checker, grammar checker, question answering, etc. Therefore, the researchers in the area of NLP application can use the design of our model or the implemented system as input or as a pre-processing component within their research.

As a future work, we would like to suggest the following key points:

- Extending this work by training the system in a large corpus and using large tagset that can identify different features such as gender, number, tense, etc.
- Preparation of a balanced corpus that contains texts which represent different genres like newspapers, fiction, textbooks, parliamentary reports, etc.
- Comparison of two hybrid approaches: the one that is done for this thesis and the hybrid of rule-based and Artificial Neural Network for Kafi-noonoo language.
- Comparative study of three different approaches (HMM based, rule-based, and Artificial Neural Network based taggers for Kafi-noonoo language with more training and testing data)

References

[1] Abbebe Asfayi, Dubbaalee Saahili, Geetaachew Kochi, Tegbaaru Mangashi, and Tewaabech Takili. *Kafi-noonoo Structure I*. Bonga College of Teacher Education, Department of Kafi-noonoo, Bonga, 2004.

[2] Abbebe Asfayi, Dubbaalee Saahili, Geetaachew Kochi, Tegbaaru Mangashi, and Tewaabech Takili. *Kafi-noonoo Structure II*. Bonga College of Teacher Education, Department of Kafi-noonoo, Bonga, 2004.

[3] Bird Steven. "NLTK: the natural language toolkit". In: *proceeding of CLOING/ACL on interactive presentation session*, Association for computational Linguistics, Morristown, NJ., Vol 1, Sydney, 2006.

[4] Charniak Eugene. *Introduction to artificial intelligence*. Addison-Wesley, Boston, 1984.

[5] Daniel Jurafsky and James Martin. *Speech and Language Processing: An Introduction to Natural Speech Recognition*. Prentice-Hall, Inc, New Jersey, 2000.

[6] Eric Brill. "A Simple Rule-Based Part-of-Speech Tagger". In: *proceeding of the third conference on Applied Natural Language Processing*, Trento, pp. 152-155, 1992.

[7] Eric Brill. "Transformation-Based Error-Driven Learning and Natural Language Processing: a Case Study in Part-of-Speech Tagging". Department of Computer Science, Association for Computational Linguistics, The Johns Hopkins University, Vol.21, No. 4, Baltimore, pp.543-565, 1995.

[8] Fahim Muhammad Hasan, Naushad UzZaman and Mumit Khan. "Comparison of Different POS Tagging Techniques (n-grams, HMM and Brill‟s Tagger) for Bangla". Center for Research on Bangla Language Processing, BRAC University, Bangladesh, pp. 4-14, December 2006.

[9] Fleming Harold. *The non-Semitic languages of Ethiopia*. African Studies: Michigan State University, Michigan, 1976.

[10] 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.

[11] Haykin Simon. *Neural Networks: A comprehensive Foundation*. Macmillan, Ontario, 1994.

[12] Helmut Schmid. "Part-of-Speech Tagging with Neural Networks". Institute of Computational Linguistics, Azenbergstr, Germany, pp. 172-176, 1994.

[13] Joakim Nivre. "Sparse data and smoothing in statistical part-of-speech tagging". *Journal of Quantitative Linguistic*, Goteborg, pp.1-17, 2000.

[14] Jackson Peter and Moulinier Isabelle. *Natural language processing for online applications: text retrieval, extraction and categorization*. Vol. 5, Stafford St., 1984.

[15] James Allen. *Natural language Understanding*. The Benjamin/Cummings Publishing company, Redwood, 1995.

[16] John Hall. "A Probabilistic Part-of-Speech Tagger with Suffix Probabilities". Master"s Thesis, School of Mathematics and System Engineering, Växjö University, Småland, April 2003.

[17] Khine Zin. "Hidden Markov Model with Rule Based Approach for Part of Speech Tagging of Myanmar Language". In: *Proceeding of the 3rd international conference on communications and information technology*, Florida, pp. 123-128, December 2009.

[18] Lawrence Rabiner. "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition". In: *Proceeding of the IEEE*, Vol. 77, NO. 2, New Jersey, February 1989.

[19] Levent Altunyurt, Zihni Orhan and Tunga Güngör. "A Composite Approach for Part of Speech Tagging in Turkish". In: *Proceeding international scientific conference computer science*, Bogaziçi University, Computer Engineering Dept., Istanbul, 2006.

[20] Mark Hepple. "Independence and Commitment: Assumptions for Rapid Training and Execution of Rule-based POS Taggers". In*: proceeding of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-200)*, Hon kong, October 2000.

[21] Mohammed Hussen. "Part-of-speech tagger for Afaan Oromo language using Transformationla Error Driven Learning (TEL) approach". Master"s Thesis, Addis Ababa University, Addis Ababa, February 2010. Unpublished

[22] Natural language processing: http://www.language.worldofcoputing.com, last visited on Aug 9, 2013.

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

[24] Python programming language: http://www.python.org, last visited on Aug 8, 2013.

[25] Richard Lippmann. "An introduction to Computing with Neural Nets". *IEEE, ASSP Magazine*, New York, pp.4-22, April 1987.

[26] Sandipan Dand, Sudeshna Sarkar, and Anupam Basu. "Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario". Department of Computer Science and Engineering Indian Institute of Technology Kharagpur, Kharagpur, 2007.

[27] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language processing with python: Analyzing text with the natural language toolkit*. O"Reilly Media, 1st ed, Cambridge, 2009.

[28] Solomon Asres. "Automatic Amharic Part-of-Speech Tagging Using Hybrid Approach (Neural Network and Rule-Based)". Master"s thesis, Addis Ababa University, Addis Ababa, 2008.

[29] Teklay Gebregzabiher. "Part-of-Speech Tagger for Tigrigna Language". Master"s Thesis, Addis Ababa University, Addis Ababa, November 2010. Unpublished

[30] Truscott John. "The effect of error correction on learners" ability to write accurately". *Journal of Second Language Writing*, Orlando, pp.255-272, December 2007.

Appendices

Appendix A: sample corpus

Untagged version

Damee nashiroo woyee beshiti damee xu'o halleebeeto damee woce gommo xebee gaatane. Ebi damee woce gommo(/blood vessels/) xebbegaata damoo bi hammemmi gommon muccemmona damee xu'o hanifiye.

Shaahiyooch xebbe kelloona kechiibeeti aaconaa gabini kelloona kechiibeeti aaconon goomiibot.Aabi oogi giidehe?Mullo damoon de'onaa imonoch bi hakkoommon shuuno qaawiihe.Shomboowaanee waabeeti damoon de'oyich biwaan beeti andire damoon daachiyoo qaawiihe.Ebi safiro kuxegaatane asho qitiye no iyaabeeto.Ebi shafiroon quyooyich mullo kufiibeeti damoo xebbe gommona bi hammimmona damoo nashiriye getteehe. Ebi xebbe gommona xu'eebee besho beshii beshii bi hamigaata damee woco neexo hakkiye; mulloonaa yeyitoona miixeyoo hakkiye; wodde aaboon udi-wuxi(silent killer) getteebeeti damee nashiroo wuxoochi bi beddahaa biiyee malletoon bekkiyoo qayo hakkiye.

Damee xuqqi oogiishoon ariyooyich ariyechina'o gutte tochee yawoona gaacheeheete.Eboshiyoo siistoolikoonaa daayastoolikoona getteeheete.Siistoolikee damee xu'o mullo damoon kuufooyich bi qodemmona tacheehe.Daayastoolike damee xu'o wotta mullo gene damoon de'oyich bi ciixemmona tacheehe.Ebi gutte tachee shaaheena'on tachiibeti tachee qiico Isfiigimomaanometiro/Sphygmomanometer getteehe.

Genje goorooch beddaahaa xure maayoon shaahi shaahoona mataanoon yagoo shaacoo getteehe.Ikee asho bedditi xure maayoon maataanoon bi yagigaata bi maggittino shappiibee hammiye, woyee dalliye. Xure maayoon mamo no neechigaata no ashittino ebiyee aaf no kiciti awuyi mooyon gooree wottich mamo hakkiye. Bedditi maayoon mami qoco shebii bi hamigaata giidee danooyich no ashittino no biddooch beeti pirootinoon tiichi mamo kottiye. Ebichi genjooch, wulle no achee qeppeena'o: mullo, qamoonaa maac meenona giishee giishee hammiyeete. Mullo giisho baach tuneyaanoon damoon gaawuchii xu'och giidoon muccehe.

Shaaceti asho bi giidoo niretaana shalligee hakkoo tuneti goorooch beddaahaa goggee beemo hakkiye. Shaacee gooroo genjee hammimmona no meeno yicciye; bari bare achee qeppena'o ebiyee aaf echeti aachereena'on yechi'i nafii nafriijjeheete; gooqqo shisheehe, aqqehe, shuuqqiye; maachaannamee danee shalligoo nirehe; baroo barooyee anaamochi ispermee qannayee hakkoo kishee kishee hammiye; maacheena'ochi dupphee shakkiyee shafiro duubehe. Shaacee goorooba xifeeti xifo kofiye, woyee boriho hakkiye; Shaacee naboona no ashittino biiyoon wushoo bi womiibeetoch meeti biiyo noon tuumehe.Ebi

iritoo waabeeto no achee maggittino balloochee 30 kishoona bi shappigaatane.

Girttino shaacee naboone bi gettetaana, bare nabeena'o maayoona irimmoon deewo hakkiyeete. Shaahiyooch biiyee naboona no maa beeti maayoon gaawuchaa yikkoon no mawigaata maayo tate gommona no ashittinooch hammache.

Shaacoona irimmeti asheena'on gaawuchii yesho woyee bixo qayigaata qitoo hakkiyeete.Shaacoon aalliyooch aabchibona?

Hajjiyee guuphina'on yechiti pirograamoon gaache hiddee shuunoonaa elektroonikee gommona danee shuunoon shuuneebeeti maashino kompiyuuteero getteehe.

Pirograamo meeti aaboon kompiyuuteeree maaacooch bee kompiyuuteeree elektroonikee qeppeen'ona shuunooch heechiye.ebi pirograamoona shuuneeti shuuneena'o kompiyuuteero baad cikkii bekkiyaache kompiyuuteeroona shuuneeti shuuneena'on aafoona be'o hakkeemmo maachi qiiceena'on getoommon, shaaoon bekkiibeeti mooniiteeroo woyee dukkee maashinoon (printers)gaachetena baachiye. kompiyuuteero michiimmi, daacaalli wodde shuuneena'on kaatee shuunehe.

Ashi bushoo kompiyuuteeroon wodde mooyon shuunooyich gaachehe.giixee shuunoochena,giiechoch gijjoonaa kechiti gijjoonon hiddii bekkiihe; gijjoon ikke xaa'oochee bare wohi xaa'ooch beshiihe. Kexoochena, gishiishi kompiyuuteere na'o kechi maaci aqqaa gaamoon quyeheete; bekkiiheete; gooroon bekkiiheete; viidiyee deekkoon qechiyeete; hicciyeete.Kaameellee maacena, nadaajjo beemonaa cimonon bekkiiheete; kaameello hammiti gommi wohonaa bi gaacheti nadaajjoonon arichiiheete.DVD'n gaache'i duubon waayiiheete. Doyoochena, dechi doyee baqqoochee tii kalkuleesooch beddaahaa dojjoo hakkiyeete; waamona waayeemmon baach tuneyaanoon aafoona ciichemmon yeshii'i kuchi maacee doyoon kette gommona giddiyooch gaacciyeete. Shaahioyooch, kaameelless shuunoonaa bi achee qeppoonon noonoona biriibee viidiyoona bekkii doyoon kettiyoo hakkeehe.Eboommon, bechee aafo collee, waamon bi kichiibeeta bekkiyoo hakkeehe.Gabini qayee shuunoochena, shinnaaawaunneeti qiheena'on boono shaahi shaahoona kotii'i meni qayechochi shuunoon kettiiheete.Kicee yibbaatoochena kompuuteero shuuneyaani shuuno aall getigaata hoxiyaache.

Kompiyuuteero gutte wulle qeppeena'on gaache'i damba no giddiiti gaaceena'on immiye.Ikkinnoo, aafoona no ciinnaabeeti kompiyuuteeree qeppoo haardweero getteehe. Guttinnoo, DVD/CD'na woyee bare gommona ikke kompiyuuteeroochee bare kompiyuuteerooch besho hakkimmi pirograamo sooftweero/Software getteehe. Haardweero no qelloona shaaheemona softweero wotta no qellooch beeti shalligoona shaahiyoo hakkeehe.Sooftweero shuuneeto ikke tuneti shuunoon shuunebe getenane.Shaahiyooch, koorichee koorooyich, fotograafoon bekkiyoonaa mayonoch, kompiyuuteeree shuunoon tachooyich, ikkikke shinnaawunneeti qiheena'on boono xaa'i xaa'oona kotiyooyich hakkimmo sooftweeroone.

Tagged version

Damee/ADJ nashiroo/N woyee/C beshiti/VC damee/ADJ xu'o/N halleebeeto/VC damee/ADJ woce/ADJ gommo/N xebee/ADJ gaatane/V ./PUNC Ebi/ADJ damee/ADJ woce/ADJ gommo/N xebbegaata/ADJC damoo/N bi/PRON hammemmi/VC gommon/NC muccemmona/VC damee/ADJ xu'o/N hanifiye/V ./PUNC

Shaahiyooch/C xebbe/ADJ kelloona/NC kechiibeeti/ADJC aaconaa/NC gabini/ADJC kelloona/NC kechiibeeti/ADJC aaconon/NC goomiibot/V ./PUNC Aabi/PRONI oogi/ADV giidehe/V ?/PUNC Mullo/N damoon/NC de'onaa/VC imonoch/VC bi/PRON hakkoommon/VC shuuno/ADV qaawiihe/V ./PUNC Shomboowaanee/NC waabeeti/VC damoon/NC de'oyich/VC biwaan/PRONPREP beeti/AUX andire/ADJ damoon/NC daachiyoo/ADV qaawiihe/V ./PUNC Ebi/PRON shafiro/NC kuxegaatane/VC asho/N qitiye/V no/PRON iyaabeeto/V ./PUNC Ebi/PRON shafiroon/NC quyooyich/VC mullo/N kufiibeeti/VC damoo/N xebbe/ADJ gommona/NC bi/PRON hammimmona/VC damoo/N nashiriye/V getteehe/V ./PUNC Ebi/PRON xebbe/ADJ gommona/NC xu'eebee/ADVC besho/V beshii/ADVC beshii/ADVC bi/PRON hamigaata/VC damee/ADJ woco/N neexo/V hakkiye/V ;/PUNC mulloonaa/NC yeyitoona/NC miixeyoo/V hakkiye/V ;/PUNC wodde/ADJ aaboon/ADVC udi-wuxi/N getteebeeti/VC damee/ADJ nashiroo/N wuxooch/VC bi/PRON beddahaa/VC biiyee/ADJ malletoon/NC bekkiyoo/V qayo/V hakkiye/V ./PUNC
Damee/ADJ xuqqi/NC oogiishoon/ADJC ariyooyich/VC ariyechina'o/NC gutte/CARDN tochee/ADJ yawoon/NC gaacheeheete/V ./PUNC Eboshiyoo/C siistoolikoonaa/NC daayastoolikoona/NC getteeheete/V ./PUNC Siistoolikee/ADJ damee/ADJ xu'o/N mullo/N damoon/NC kuufooyich/VC bi/PRON qodemmona/VC tacheehe/V ./PUNC Daayastoolike/ADJ damee/ADJ

Genje/ADJ goorooch/NC beddaahaa/VC xure/ADJ maayoon/NC shaahi-shaahoona/ADV mataanoon/VC yagoo/V shaacoo/N getteehe/V ./PUNC Ikke/CARDN asho/N bedditi/VC xure/ADJ maayoon/NC maataanoon/VC bi/PRON yagigaata/VC bi/PRON maggittino/ADJ shappiibee/ADV hammiye/V ,/PUNC woyee/C dalliye/V ./PUNC Xure/ADJ maayoon/NC mamo/V no/PRON neechigaata/VC no/PRON ashittino/N ebiyee/PREP aaf/PREP no/PRON kiciti/VC awuyi/ADJ mooyon/NC gooree-wottich/ADV mamo/V hakkiye/V ./PUNC Bedditi/ADJ maayoon/NC mami/ADV qoco/N shebii/ADV bi/PRON hamigaata/VC giidee/ADV danooyich/VC no/PRON ashittino/N no/PRON biddooch/NC beeti/VC pirootinoon/NC tiichi/VC mamo/V kottiye/V ./PUNC Ebichi/PRONC genjooch/ADJ ,/PUNC wulle/ADJ no/PRON achee/ADJ qeppeena'o/NC :/PUNC mullo/N ,/PUNC qamoonaa/NC maac/ADJ meenona/NC giishee/ADJ giishee/ADJ hammiyeete/V ./PUNC
Mullo/N giisho/ADJ baach/C tuneyaanoon/VC damoon/NC gaawuchii/ADJ xu'och/VC giidoon/VC muccehe/V ./PUNC

Shaaceti/ADJ asho/N bi/PRON giidoo/V niretaana/VC shalligee/NC hakkoo/V tuneti/VC goorooch/NC beddaahaa/VC goggee/ADV beemo/V hakkiye/V ./PUNC Shaacee/ADJ gooroo/N genjee/ADJ hammimmona/VC no/PRON meeno/N yicciye/V ;/PUNC baribare/ADJ achee/ADJ qeppena'o/NC

ebiyee/PRON aaf/PRON echeti/VC aachereena'on/NC yechi'i/VC nafii/ADJ nafriijjeheete/V ;/PUNC gooqqo/N shisheehe/V ,/PUNC aqqehe/V ,/PUNC shuuqqiye/V ;/PUNC maachaannamee/NC danee/ADV shalligoo/V nirehe/V ;/PUNC baroo/ADV barooyee/ADVC anaamochi/NC ispermee/NC qannayee/VC hakkoo/V kishee/ADV kishee/ADV hammiye/V ;/PUNC maacheena'ochi/NC dupphee/NC shakkiyee/VC shafiro/N duubehe/V ./PUNC Shaacee/ADJ goorooba/NC xifeeti/ADJ xifo/N kofiye/V ,/PUNC woyee/C boriho/V hakkiye/V ;/PUNC Shaacee/ADJ naboona/ADV no/PRON ashittino/N biiyoon/NC wushoo/V bi/PRON womiibeetoch/VC meeti/ADJ biiyo/N noon/PRON tuumehe/V ./PUNC Ebi/PRON iritoo/N waabeeto/VC no/PRON ache/ADJ maggittino/VC balloochee/NC 30/CARDN kishoona/NC bi/PRON shappigaatane/V ./PUNC

Girttino/N shaacee/ADJ naboone/V bi/PRON gettetaana/VC ,/PUNC bare/ADJ nabeena'o/NC maayoona/NC irimmoon/ADJC deewo/V hakkiyeete/V ./PUNC Shaahiyooch/NC biiyee/NC naboona/NC no/PRON maabeeti/VC maayoon/NC gaawuchaa/ADV yikkoon/VC no/PRON mawigaata/VC maayo/N tate/ADJ gommona/VC no/PRON ashittinooch/NC hammache/V ./PUNC

Shaacoona/NC irimmeti/ADJ asheena'on/NC gaawuchii/ADJ yesho/V woyee/C bixo/V qayigaata/VC qitoo/N hakkiyeete/N ./PUNC Shaacoon/NC aalliyooch/VC aabchibona/V ?/PUNC

Hajjiyee/VC guuphina'on/NC yechiti/VC pirograamoon/NC gaache/VC hiddee/NC shuunoonaa/NC elektroonikee/NC gommona/NC danee/VC shuunoon/NC shuuneebeeti/VC maashino/N kompiyuuteero/N getteehe/V ./PUNC

Pirograamo/N meeti/ADJ aaboon/ADJ kompiyuuteeree/NC maaacooch/PREP bee/V kompiyuuteeree/NC elektroonikee/NC qeppeen'ona/NC shuunooch/NC heechiye/V ./PUNC ebi/PRON pirograamoona/NC shuuneeti/VC shuuneena'o/NC kompiyuuteero/N baad/VC cikkii/VC bekkiyaache/V kompiyuuteeroona/NC shuuneeti/VC shuuneena'on/NC aafoona/NC be'o/V hakkeemmo/V maachi/ADJ qiiceena'on/NC getoommon/VC ,/PUNC shaaâ€™oon/VC bekkiibeeti/VC mooniiteeroo/N woyee/C dukkee/ADJ maashinoon/NC gaachetena/VC baachiye/V ./PUNC kompiyuuteero/N michiimmi/VC, daacaalli/VC wodde/ADJ shuuneena'on/NC kaatee/ADV shuunehe/V ./PUNC

Ashi/ADJ bushoo/N kompiyuuteeroon/NC wodde/ADJ mooyon/NC shuunooyich/NC gaachehe/V ./PUNC giixee/NC shuunoochena/NC ,/PUNC giice/NC echoch/VC gijjoonaa/NC kechiti/VC gijjoonon/NC hiddii/ADV bekkiihe/V ;/PUNC gijjoon/NC ikke/CARDN xaa'oochee/NC bare/ADJ wohi/ADJ xaa'ooch/NC beshiihe/V ./PUNC Kexoochena/NC ,/PUNC gishiishi/ADJ kompiyuuteerena'o/NC kechi/ADJ maaci/PREPC aqqaa-gaamoon/ADJC quyeheete/V ;/PUNC bekkiiheete/V ;/PUNC gooroon/NC bekkiiheete/V ;/PUNC viidiyee/NC deekkoon/NC qechiyeete/V ;/PUNC hicciyeete/V ./PUNC

Kaameellee/NC maacena/PREP ,/PUNC nadaajjo/N beemonaa/VC cimonon/VC bekkiiheete/V ;/PUNC kaameello/N hammiti/VC gommi/ADJ wohonaa/NC bi/PRON gaacheti/VC nadaajjoonon/NC arichiiheete/V ./PUNC DVD'n/NC gaache'i/VC duubon/NC waayiiheete/V ./PUNC Doyoochena/NC ,/PUNC dechi/PREP doyee/NC baqqoochee/NC tii/VC kalkuleesooch/NC beddaahaa/VC dojjoo/N hakkiyeete/V ;/PUNC waamona/NC waayeemmon/VC baach/C tuneyaanoon/VC aafoona/NC ciichemmon/VC yeshii'i/VC kuchi/NC maacee/ADJ doyoon/NC kette/ADJ gommona/VC giddiyooch/VC gaacciyeete/V ./PUNC Shaahioyooch/NC ,/PUNC kaameelle/NC shuunoonaa/VC bi/PRON ache/ADJ qeppoonon/NC noonoona/NC biriibee/VC viidiyoona/NC bekkii/VC doyoon/NC kettiyoo/ADV hakkeehe/V ./PUNC Eboommon/C ,/PUNC bechee/ADJ aafo/N collee/V ,/PUNC waamon/NC bi/PRON kichiibeeta/VC bekkiyoo/VC hakkeehe/V ./PUNC Gabini/ADJ qayee/NC shuunoochena/NC ,/PUNC shinnaaawaunneeti/ADJ qiheena'on/NC boono/PRON shaahi-shaahoona/ADJ kotii'i/V meni/PRON qayechochi/NC shuunoon/NC kettiiheete/V ./PUNC Kicee/ADJ yibbaatoochena/NC kompuuteero/N shuuneyaani/VC shuunoo/N aalla/V getigaata/VC hoxiyaache/V ./PUNC

Kompiyuuteero/N gutte/CARDN wulle/ADJ qeppeena'on/NC gaache'i/VC damba/PREP no/PRON giddiiti/VC gaaceena'on/VC immiye/V ./PUNC Ikkinnoo/ORDN ,/PUNC aafoona/NC no/PRON ciinnaabeeti/VC kompiyuuteeree/NC qeppoo/N haardweero/N getteehe/V ./PUNC Guttinnoo/ORDIN ,/PUNC DVD/N CD'na/NC woyee/C bare/ADJ gommona/NC ikke/CARDN kompiyuuteeroochee/NC bare/ADJ kompiyuuteerooch/NC besho/VC hakkimmi/VC pirograamo/N sooftweero/N getteehe/V ./PUNC Haardweero/N no/PRON qelloona/NC shaaheemona/VC softweero/N wotta/C no/PRON qellooch/NC beeti/VC shalligoona/NC shaahiyoo/ADV hakkeehe/V ./PUNC Sooftweero/N shuuneeto/V ikke/CARD tuneti/VC shuunoon/NC shuunebe/ADV getenane/V ./PUNC Shaahiyooch/NC ,/PUNC koorichee/NC koorooyich/NC ,/PUNC fotograafoon/NC bekkiyoonaa/VC mayonoch/VC ,/PUNC kompiyuuteeree/NC shuunoon/NC tachooyich/VC ,/PUNC ikkikke/ADJ shinnaawunneeti/ADJ qiheena'on/NC boono/PRON xaa'i-xaa'oona/ADV kotiyooyich/VC hakkimmo/VC sooftweeroone/V ./PUNC

Appendix B: Brill tagger learned rules

Sample lexical rules

ADV ->ADJ if the text of words i-3…i-1 is „Ne"

ADV ->ADJ if the text of the following word is „maacoona"

N -> ADJ if the text of the preceding word is „Gundo" and the text of the following word is „battoo"

VC ->V if the text of the preceding word is „guuti", and the text of the following word is „iye"

VC ->ADV if the text of words i-2…i-1 is „ciinaa"

……..

Sample contextual rules

PRONI ->VC if the tag of the preceding word is „N" and the tag of the following word is „VC"

V ->PRON if the tag of the preceding word is „ADJ" and the tag of the following word is „VC"

PRON -> C if the tag of the following word is „PUNC"

V ->ADJ if the tag of the preceding word is „ADJ" and the tag of the following word is „N"

NPREP -> NC if the tag of the preceding word is „VC" and the tag of the following word is „PRON"

……