ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

NAMED ENTITY RECOGNITION FOR AMHARIC LANGUAGE

BY: MOGES AHMED MEHAMED

ADVISOR: SEBSIBE H/MARIAM (PhD)

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

November, 2010

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

NAMED ENTITY RECOGNITION FOR AMHARIC LANGUAGE

BY: MOGES AHMED MEHAMED

ADVISOR: SEBSIBE H/MARIAM (PhD)

APPROVED BY

EXAMINING BOARD:

1. Dr. SEBSIBE H/MARIAM, Advisor _____
2. _____ _____
3. _____ _____

Acknowledgements

First and foremost, to God, who makes everything possible. I would also like to extend my deepest gratitude to my advisor Dr.Sebsibe H/Mariam for his valuable time, guidance and understanding through this thesis. My advisor provided helpful and constructive ideas, comments and experience which enabled me to gain good research experience on computer science.

Thank you Dr.Sebsibe!

My heartfelt gratitude goes to Dr. Erick Breck and Bob Carpenter for your support when I have a problem on LingPipe tool used for my thesis. Thanks a lot!!!!!!!

My special thanks goes to all my classmates especially, Tesfaye Guta, Seada Ali, Mequanint Munye and Teklay G/Her (First year group members for projects), Mandefro Legese for your time for discussion on LingPipe tool.

And last but not the least, my indebtedness goes to my family and to Tigist Mulugeta for your moral support.

Table of Contents

-

List of Tables

List of Figures

Abbreviations

ACE	Automatic Content Extraction.
AI	Artificial Intelligence.
ANERcorp	Arabic named Entity Recognition Corpus.
ANER	Amharic Named Entity Recognition.
ANERsys	Arabic Named Entity Recognition system.
CoNLL2002	Conference on Computational Natural Language Learning 2002
CR	Co-reference Resolution.
CRFs	Conditional Random Fields.
ERTA	Ethiopian Radio and Television Agency
HMM	Hidden Markov Models.
I	Inner.
IE	Information Extraction.
IES	Information Extraction System.
IJCNLP	International Joint Conference on Natural Language Processing.
IR	Information Retrieval.
ISCII	Indian Script Code for Information Interchange.
K	Kilo (1024).
LERC -UoH	Language Engineering Research Centre at the Department of Computer and Information Sciences, University of Hyderabad.
LOC	Location.
Ltd	Limited.
ME	Maximum Entropy.
MT	Machine Translation.
MUC	Message Understanding Conference.
NE	Named Entity.
NER	Named Entity recognition.
NERAL	Named Entity recognition for Amharic Language.
NERS	Named Entity Recognition System.
NLP	Natural Language Processing.
ORG	Organization.
PER	Person.
POS	Part Of Speech.
SSL	Semi-Supervised Learning.
STP	Scenario Template Production.
SVM	Support Vector Machines.
TRC	Template Relation Construction.
TEC	Template Entity Construction.
US	United States.
UTF	Unicode Transformation Format.
WIC	Walta Information Center.
WWW	World Wide Web.

Abstract

Named Entity Recognition (NER) is a process of identifying and categorizing all named entities in a document into predefined classes like person, organization, location, time, and numeral expressions. This identification and classification of proper names in text has recently considered as a major importance in natural language processing as it plays a significant role in various types of NLP applications, especially in information extraction, information retrieval, machine translation, and question-answering. This paper reports about the development of a NER system for Amharic using Conditional Random Fields (CRFs). Though this state of the art machine learning method has been widely applied to NER in several well-studied languages, this is the first attempt to use this method to Amharic language.

The system makes use of different features such as word and tag context features, part of speech tags of tokens, prefix and suffix. Since feature selection plays a crucial role in CRF framework, experiments were carried out to find out most suitable features for Amharic NE tagging task. During the experiment, four different scenarios were considered based on the different combination of features. In the first scenario all the features were considered, in the second scenario all the features except POS tags of tokens were considered. In the third and fourth scenarios all the features except prefix and suffix respectively were considered.

The experimental results show that for different combinations of features, we have got different results. In scenario one experiment, we have got Precision, Recall and F-measure of 72%, 75% and 73.47% respectively. Taking this as a base line we made the remaining experiments. The remaining experiments on scenario two, three and fourth, its F-measure of 69.70%, 74.61%, and 70.65% respectively were obtained.

From the above results, it is possible to make a conclusion that word context features, POS tags of tokens and suffix are important features in NE recognition and classification for Amharic text.

Keywords: Named Entity Recognition, Conditional Random fields, Named Entities, Amharic Named Entity Recognition.

Chapter One

Introduction

1.1 Background

Named Entity Recognition (NER) is a process of identifying and categorizing all named entities in a document into predefined classes like person, organization, location, time, and numeral expressions. It can also be treated as a two step process i.e. identification of proper nouns and its classification. Identification is concerned with marking the presence of a word / phrase as Named Entity (NE) in the given sentences and classification is for denoting role of the identified NE [29].

Proper identification and classification of NEs are very crucial and create very big challenges to Natural Language processing (NLP) researchers. The level of ambiguity in NER makes it difficult to attain human performance. This problem of correct identification of NEs is specifically addressed and bench marked by the developers of Information Extraction System (IES) [2]. In addition to this, the challenge for NER from language to language is different. For example, in English NEs usually starts with capital letter as a result it makes easier to recognize names in English. But in Amharic, there is no such rule. As a result, NER in Amharic language is becoming difficult and challenging in comparison to English.

This identification and classification of proper names in text has recently considered as a major importance in natural language processing as it plays a significant role in various types of NLP applications, especially in information extraction, information retrieval, machine translation, and question-answering [20]. In this case, it is important to discuss about natural language processing and its applications to clearly see the importance of NER in NLP applications.

The term natural language processing encompasses a broad set of techniques for automated generation, manipulation and analysis of natural or human languages. Although most NLP techniques inherit largely from linguistics and artificial Intelligence, they are also influenced by relatively newer areas such as machine learning, computational statistics and cognitive science [7]. NLP is the use of computers to process written and/or spoken language for some practical,

useful purpose. Its goal is to design and build software that will analyze, understand, and generate languages that humans use naturally, so that eventually we will be able to address our computer as we were addressing another person. Some of the application areas of NLP include: information retrieval, information extraction, speech recognition, machine translation, question answering, automatic text or document summarization, co-referencing, dialog system management [2,5,6].

Information Retrieval (IR) is the science of searching for a document from collection of documents, such as the World Wide Web (WWW). Automated information retrieval systems are used to reduce what has been called information overload. Many universities and public libraries use IR systems to provide access to books, journals and other documents. Web search engines are the most visible IR application [13]. In this case, identifying named entities from the users query has a great contribution in retrieving the required information since the valuable information in text is usually located around proper names [20].

Information Extraction (IE) is a more recent application area of NLP. It extracts structured data or knowledge from un-structured text by identifying references to named entities as well as stated relationships between such entities. In this case, named entity recognizer is part of this task [27].

Machine Translation (MT), another application area of NLP, which is the use of computers to automate some or all of the process of translating document written in one language into another language. MT requires a deep and rich understanding of the source language and the input text, and a sophisticated creative command of the target language [14].

Text or document summarization is the higher levels of NLP that can allow an implementation that reduces a larger text into a shorter which constituted abbreviated narrative representation of the original document. In summarization of texts/documents, named entities are useful to find key expressions; as a result we need a NE recognizer for text or document summarization [14, 15].

Co-referencing is the process of identifying terms or information objects that are referring to the same object in the real world [16]. For example, if we have the sentence

“Mr. John has explained that the source of the problem for the conflict between the two countries is Blue Nile River. He also mentioned the solution how the conflict can be resolved. ”

In this case, if the main focus of the system is to resolve for what the pronoun “He” refers to in the sentence, there should be a NER system that recognizes the name “Mr.John” in the first sentence before co-reference resolution task.

A dialog system is a computer system intended to converse with a human, with a coherent structure. Dialog systems have employed text, speech, graphics, gestures and other modes for communication on both the input and output channel. [17].

As we can see from the above applications of NLP, NER is one of the components to be in place so as to successfully develop those applications [1, 2, 5].

1.2 Statement of the Problems

First and foremost the reason why NER for Amharic is chosen as a research agenda is due to its crucial relevance to work on most researches in the area of NLP for Amharic. Moreover, there is no any attempt on NER for Amharic so far. Due to this, it is a big challenge for researchers that work on Amharic natural language processing applications that need named entity recognizer. Mostly they used gazetteers i.e. list of names of persons, locations, organizations etc in file(s). However, named entities are too numerous and are constantly evolving. Even when named entities are listed in the dictionaries, it is not always easy to decide their senses. There can be semantic ambiguities [2]. For example, the Amharic word “አባይ /Abay/” refers to person name, place or organization name. The study will address those problems and develops named entity recognizer for Amharic that does both identification and classification.

1.3 Objective

General Objective

The general objective of this research work is to design and implement a Named Entity Recognition model for Amharic so as to improve or create enabling atmosphere for natural language processing tasks.

Specific objective

- To study features of Amharic language and issues in NEs of the language.

- To study techniques of identifying proper nouns from collection of nouns for Amharic language.
- To review existing approaches on NER for various languages.
- To design and develop a model for NER system for Amharic language.
- To develop a prototype for NER system for Amharic language.
- To test the prototype performance.

1.4 Methods and Techniques

In researching this NER for Amharic, the study used different methods and techniques such as literature review, data collection, model selection, identifying development tools and performance analysis.

1.4.1 Literature review: journals, papers and books about NLP, Amharic language structure and characteristics, approaches that are used so far in NER models, applications and role of NER system is reviewed.

1.4.2 Data collection: Walta Information Center (WIC) - Tagged Amharic News Corpus is used for training and testing purpose. The corpus contains 1,065 Amharic news articles (210,000 words) from the Walta Information Center (http://www.waltainfo.com/). The news articles span the period from 1998 - 2002 and have been tagged for part of speech and punctuation [18].

1.4.3 Data preparation: For this study, data (sentences) are taken from WIC Amharic news corpus that contains at least one named entity and then each words of the sentences are tagged for named entity tags such as B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG and O

Where,

- B-PER is beginning of person named entity,
- I-PER is inner of person named entity,
- B-LOC is beginning of location named entity,
- I-LOC is inner of location named entity,

- B-ORG is beginning of organization named entity, and
- I-ORG is inner of organization named entity.

To train part of speech tagger, sentences are taken from WIC corpus. Tokens of those sentences were already tagged with their respective part of speech tags.

1.4.4 Development tools: LingPipe is used as a tool for developing the NER for Amharic language. *LingPipe* is a suite of Java libraries for the linguistic analysis of human language and developed by *Alias-I*. It is by default prepared for the detection and the classification of NEs such as persons, organizations and locations in the English language, but it is also possible to customize it for other languages. It is an open-source and free of charge tool for research purpose [3].

1.4.5 Model selection: Widely used approaches for NER are statistical machine learning techniques, ruled based system or hybrid approach [29]. In this research one of the statistical approaches called Condition Random Fields (CRFs) will be used since it is the current state of the art. For POS tagger, HMM model is used as the model in LingPipe uses HMM approach.

1.4.6 Feature set: Set of features that will be applied to Amharic NER (ANER) task, are context word feature (Previous and next words of a particular word), Word suffix and prefix of the current, previous and/or next token, the Part Of Speech (POS) tags of the current and/or the surrounding word(s) will be used as features.

1.4.7 Performance Analysis: Performance analysis of the proposed system will be analyzed using Precision, Recall and F-measure test on various types of sufficiently large test samples.

1.5 Scope and Limitation

Named entities have a number of categories like organization, location, person, numeric values like numbers, date, and percentage. In addition, some of the entities are domain specific. Moreover, within each category there could be additional sub categories. For example, for location may have city, mountain, river, sea, village etc. as its subcategory. But in this study, due to time constraints it focuses only to name entities that belongs to Organization (ORG), Location (LOC), and Person (PER) without considering the subcategories.

1.6 organization of the work

The rest of this thesis is organized as follows. In Chapter 2, background information of named entity recognition is described. The Chapter explains different types of approaches to named entity recognition specifically it explains HMM and CRFs based approaches. Chapter 3 critically reviews related works for named entity recognition. Chapter 4 presents design and implementation of ANER system using conditional random fields approach. Chapter 5 presents the experimental results of the proposed system along with its discussion. Finally, Chapter 6 concludes the thesis with the research findings and future works.

Chapter Two

Literature Review

2.1 Natural Language Processing

The amount of natural language text that is available in electronic form is increasing every day. However, the complexity of natural language can make it very difficult to access the information in that text. The state of the art in NLP is still a long way from being able to build general-purpose representations of meaning from unrestricted text [36]. NLP is theoretically motivated range of computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications [6].

To clarify this, let's see some elements of the above definition for NLP. Firstly, the notions for "range of computational techniques" indicates as there are multiple methods or techniques from which to choose to accomplish a particular type of language analysis. The phrase "naturally occurring texts" indicates, for any language in written or oral form, must be put in the way humans used to communicate to one another .i.e. the text being analysed should not be specifically constructed for the purpose of the analysis. The phrase "levels of linguistic analysis" refers to the fact that there are multiple types of language processing levels when humans produce or comprehend language. Humans normally utilize all of those levels since each level conveys different types of meaning. But various NLP systems utilize different levels or combination of levels of linguistic analysis and this is seen in the differences amongst various NLP applications. The phrase "human-like language processing" reveals that NLP is considered as a discipline with in Artificial Intelligence (AI) since it strives for human-like performance [6].

Based on [24], at the core of any NLP task there is an important issue of natural language understanding. In order to understand natural languages, it is important to be able to distinguish

among the following seven interdependent levels that people use to extract meaning from text or spoken languages:

- Phonetic or phonological level that deals with pronunciation,
- Morphological level that deals with the smallest parts of words, that carries a meaning, and suffixes and prefixes,
- Lexical level that deals with lexical meaning of words and parts of speech analyses,
- Syntactic level that deals with grammar and structure of sentences,
- Semantic level determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence,
- Discourse level that deals with the structure of different kinds of text using document structures or (While syntax and semantics work with sentence-length units, the discourse level of NLP works with units of text longer than a sentence) and
- Pragmatic level that deals with the knowledge that comes from the outside world, i.e., outside the contents of the document.

A natural language processing system may involve all or some of these levels of analysis. But this study relates to lexical level of language by focusing on recognizing named entity texts [6,24]. One of the sub discipline of NLP that require NER as its major component is Information Extraction (IE). IE will be introduced in the next section as researchers admit the main role of NER is for IE purpose.

2.2 Information Extraction

With the increasing volume of publicly available information, companies need to develop processes for mining information that may be vital for their business. Unfortunately, much of this information is presented in the form of unstructured or semi-structured texts. Software tools are not able to analyze such texts and humans would take so much time to perform this task that the information would become obsolete by the time it was available. To deal with this problem information extraction emerged as a solution [25].

According to [27], IE concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from unstructured text [25, 27]. The process takes texts

as input and produces fixed-format, unambiguous data as output. This data may be used directly for display to users, or may be stored in a database or spreadsheet for later analysis, or may be used for indexing purposes in IR applications such as Internet search engines like Google [28].
Since information extraction activity could be very complex, it is very important to decompose it in to a number of tasks. The work in [30] based on Message Understanding Conference-7(MUC-7), five tasks have been defined with in a general tasks of IE: NER (Finds and classifies names, places, etc.), Co-reference Resolution (CR)(Identifies relations between entities), Template Element Construction (TEC)(Determining the different attributes of a given entity), Template Relation Construction (TRC)(Finds relations between TE entities), Scenario Template Production (STP)(Fits TEC and TRC results into specified event scenarios) [30]. For example, if we have the statement:

"*The shiny red rocket was fired on Tuesday. It is the brainchild of Dr. Big Head. Dr. Head is a staff scientist at We Build Rockets Inc."*

NE discovers that the entities present are the rocket, Tuesday, Dr. Head and We Build Rockets Inc. CR discovers that "it" refers to the rocket. TEC discovers that the rocket is shiny red and that it is Head's brainchild. TRC discovers that Dr. Head works for We Build Rockets Inc. STP discovers that there was a rocket launching event in which the various entities were involved [28].
Thus, NER system is one of the most important preprocessing tasks for the other sub tasks of information extraction.

2.2.1 Named Entity Recognition

In comparison with other sub tasks, simplest and most reliable IE technology is NER. The term named entity recognition was originally introduced in MUC-6 in 1995. Throughout the MUC series, the term named entity came to include seven categories; persons, organizations, locations (usually referred to as ENAMEX), temporal expressions, dates (TIMEX), percentages, and monetary expressions (NUMEX). Research on named entity recognition has been carried out for a number of languages such as English, German, Spanish, Dutch, Swedish, Telugu, Bengali etc on various dataset.

The data sets used often by researchers for NER consists of news wire texts, transcribed broadcast data, or scientific texts [19]. Some examples are: LERC-UoH (Language Engineering Research Centre at the Department of Computer and Information Sciences, University of Hyderabad) Telugu corpus which contains wide variety of books and articles with a size of nearly 40Million words; Bengali news corpus developed from the archive of a widely read Bengali newspaper. The corpus contains around 34 million word forms in ISCII (Indian Script Code for Information Interchange) and UTF-8 format etc.

Figure 2.1 illustrates a screen shot of Gate NER system for English language text. The NE recognizer using the IE software distributed with the GATE system Cunningham (2002), the results shown in an output Figure 2.2 **[**28].

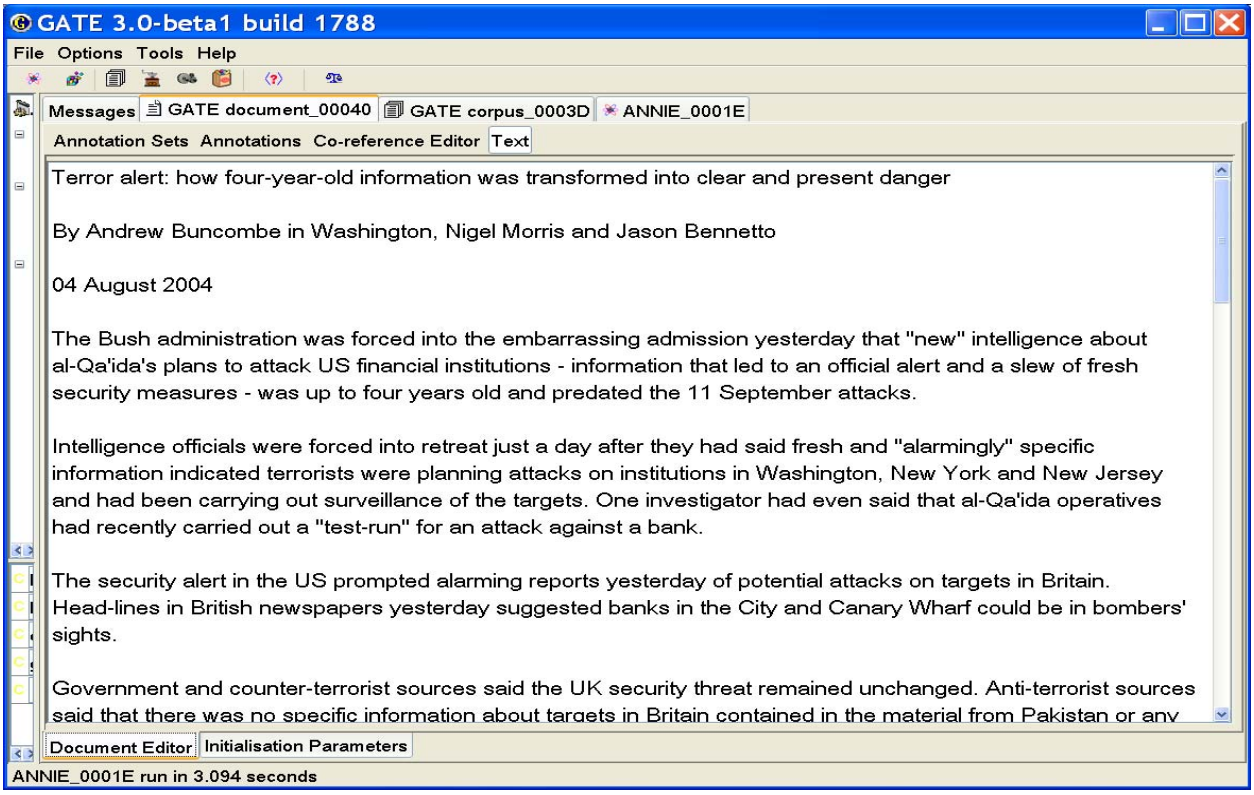

Figure 2.1 Gate NER system for English language with an input text.

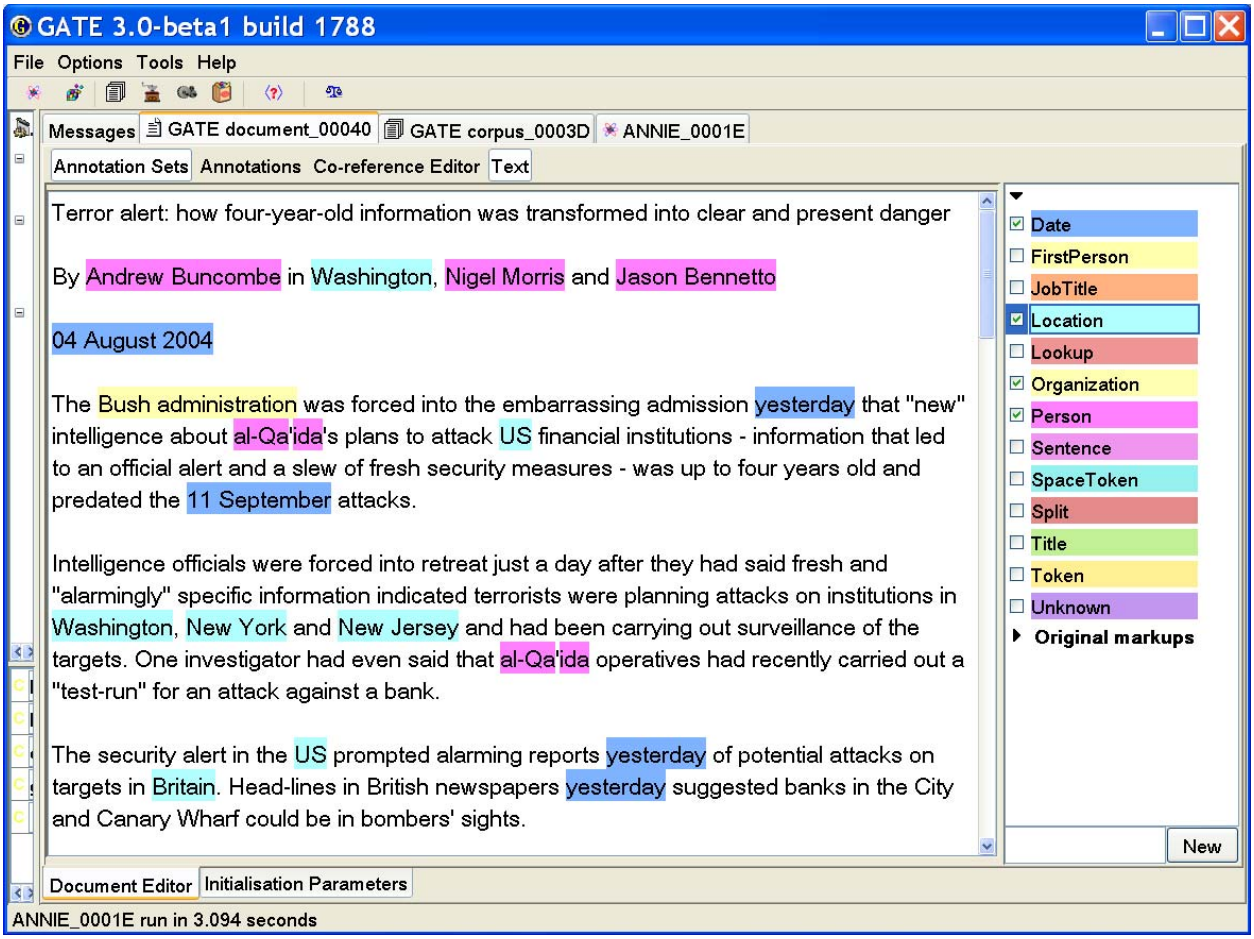

Figure 2.2 Output of Gate NER system for English texts.

The first systems for recognizing names were based on pattern matching rules and pre-compiled lists of information, but the research community has moved towards employing machine learning methods for creating such systems. The next section deals with the current and previously used approaches to named entity recognition.

2.3 Approaches to NER

There has been a considerable amount of work on NER in different languages such as English, Indian languages, Chinese etc. Much of the previous works to find names from given texts is based on one of the following approaches: (i) hand-crafted or automatically acquired rules or finite state patterns, (ii) look up from large name lists or other specialized resources and (iii) supervised machine learning approaches exploiting the statistical properties of the language [26].

The earliest work in named-entity recognition involved hand-crafted rules based on pattern matching. For instance, a sequence of capitalized words ending in ”Inc.” is typically the name of an organization in English. So, one could implement a rule to that effect. Another example of such a rule is, first name of an entity must be capitalized. The rule based approach in general uses morphological and contextual evidence of a natural language and consequently determines the named entities. This eventually leads to formation of some language specific rules for identifying named entities. Developing and maintaining rules and dictionaries is a costly affair and adaptation to different domains is difficult [26, 21].

In the second approach, the NER system recognizes only the named entities stored in its lists also called gazetteers. This approach is simple, fast, almost language independent and easy to re-target - just recreate the lists. However, named entities are too numerous and are constantly evolving. Even when named entities are listed in the dictionaries, it is not always easy to decide their senses. There can be semantic ambiguities. For example, “ፀሃይ /Tsehay/” refers to both person name as well as place name [26].

The machine learning techniques are relatively independent of language and domain and minimal expert knowledge is needed. There has been a lot of work on NER for English employing the machine learning techniques, using both supervised learning, and semi-supervised learning. Supervised approaches can achieve good performance when a large amount of high quality training data is available. However, developing large scale, high quality training data itself is a costly affair. The unavailability of such resources and the prohibitive cost of creating them lead to a method called semi-supervised learning (SSL). Semi-supervised learning approach is relatively recent and involves a small degree of supervision, such as a set of seeds, for starting the learning process [21, 23, and 26].

Supervised learning techniques include Hidden Markov Models (HMM), Decision Trees, Maximum Entropy Models (ME), Support Vector Machines (SVM), and Conditional Random Fields (CRFs). These are all variants of the supervised learning approaches that typically consist of a system that reads a large annotated corpus, memorizes lists of entities, and creates disambiguation rules based on discriminative features [23]. This research utilizes CRFs approach which extends the HMM based techniques. Under this section both the HMM and CRFs approaches will be discussed.

2.3.1 Hidden Markov Model (HMM)

One approach for modeling linear sequence structures, as can be found in natural language text, is the HMM approach. For the sake of complexity reduction, usually strong independence assumptions between the observation variables are made in HMM. This impairs the accuracy of the model. Avoiding this independence is theoretically possible but practically impossible as it is difficult to collect sufficiently large training data set.

To predict a sequence of class variables $\vec{y} = (y_1, \ldots, y_n)$ for an observation sequence $\vec{x} = (x_1, \ldots x_n)$, a simple sequence model can be formulated as a product of the prior and likely hood probabilities as shown in equation (1) [31].

$$P(\vec{y}, \vec{x}) = \prod_{i=1}^{n} p(y_i) \cdot p(x_i|y_i) \quad (1).$$

Prior probability

Likely hood probability

Dependencies between single sequence positions are not taken into account. Note that there is only one feature at each sequence position, namely the identity of the respective observation.

As shown in this equation, every classes and every observations are mutually independent. Each observation x_i depends only on the class variable y_i at the respective sequence position. However, most HMM models that are in common use allow class dependency. The modified HMM with class label dependency becomes [31]:

$$P(\vec{y}, \vec{x}) = \prod_{i=0}^{n} p(y_i|y_{i-1}) \cdot p(x_i|y_i) \quad (2).$$

Dependencies between output class variables $\vec{y}$ are modeled using training data in a supervised learning way. CRFs address exactly the same problem in a more complex and powerful approach.

2.3.2 Conditional Random Fields Model

CRFs is probabilistic model for computing the probability (i.e. p ($\vec{y}|\ \vec{x}$)) of a possible output sequence $\vec{y}$ = ($y_{1, \ldots,}\ y_n$) ε Y, given the input sequence $\vec{x}$ = ($x_1, \ldots, x_n$) ε X which is also called the observation. Here Y is the set of all possible output tag sequences and X is the set of all possible sequences of observations. Under this section a special form of CRFs which is Linear-chain CRFs is discussed in detail which is used in our NE model.

Linear chain CRFs is a special form of CRFs, which is structured as a linear chain that models the output variables as a sequence. Linear-chain CRFs can be formulated as [31]:

$$P_{\vec{\lambda}}(\vec{y}|\vec{x}) = \frac{1}{Z_{\vec{\lambda}}(\vec{x})}.exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i\, f_i(y_{j-1}, y_j, \vec{x}, j)\right) \quad (3).$$

Where n indicates the length of the sentence, m indicates the number of feature templates, λ_i represent the weights assigned to the different features in the training phase and $Z_{\vec{\lambda}}(\vec{x})$ is a normalization factor that make the probability in the range [0,1], which can be expressed as [31]:

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y}\,\epsilon Y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i\, f_i(y_{j-1}, y_j, \vec{x}, j)\right) \quad (4).$$

As we can see above, CRFs approach uses a feature function. Here it is important to discus about feature functions in the next section to clearly understand how the features are represented for the given function.

2.3.2.3 Feature Functions

The feature functions are the key components of CRFs. In our special case of linear-chain CRFs, the general form of a feature function is $\mathbf{f_i(y_{j-1}, y_j, \vec{x}, j)}$, which looks at a pair of adjacent states $\mathbf{y_{j-1}, y_j}$, the whole input sequence $\vec{x}$, and where we are in the sequence (j). These are arbitrary functions that produce a real value.

For example, we can define a simple feature function which produces binary values as,

$$f_1(y_{j-1}, y_j, \vec{x}, j) = \begin{cases} 1 & if\ y_j = PERSON\ and\ x_j = selama \\ 0 & otherwise \end{cases} \quad (5).$$

This is to mean "if the j^{th} word is selama and having a tag PERSON, then f_1 is one other wise f_1 is zero".

The usage of this feature depends on its corresponding weight λ_1. If $\lambda_1 > 0$, whenever f_1 is active (i.e. we see the word selama in the sentence and we assign it tag PERSON), it increases the probability of the tag sequence $y_{1:n}$ ($\vec{y}$).This is another way of saying the CRFs model should prefer the tag PERSON for the word selama. On the other hand $\lambda_1 < 0$, the CRF model will try to avoid the tag PERSON for selama. In this case the value of λ_1 will be learned from the corpus.

As another example, consider,

$$f_2(y_{j-1}, y_j, \vec{x}, j) = \begin{cases} 1 & if\ y_j = PERSON\ and\ x_{j-1} = ato \\ 0 & otherwise \end{cases} \quad (6).$$

This feature is active if the current tag is PERSON and the previous word is "ato". One would therefore expect a positive λ_2 to go with the feature. Note both f_1 and f_2 can be active for sentences like "ato selama…" at different j values. This example of overlapping features boots up the belief of y_2 = PERSON to $\lambda_1+\lambda_2$. This is something HMMs cannot do. HMMs cannot look at the next word, nor can they use overlapping features.

The next feature example considers state transition,

$$f_3(y_{j-1}, y_j, \vec{x}, j) = \begin{cases} 1 & if\ y_{j-1} = Other\ and\ y_j = PERSON \\ 0 & otherwise \end{cases} \quad (7).$$

This feature is active if we see the particular tag transition (OTHER, PERSON). Note it is the value of λ_3 that actually specifies the equivalent of transition probability from OTHER to PERSON [32].

In a linear chain CRFs there are two problems that have to be addressed:

Problem I: given observation x and a CRF M: How to find the most probably fitting label sequence $\vec{y}$? This problem is the most common application of a conditional random field to find a label sequence for an observation.

Problem II: given label sequences Y and observation sequences X: How to find parameters of a CRF M to maximize P ($\vec{y} \mid \vec{x}$, M)? This problem is question of how to train to adjust the parameters of M which are especially the feature weights $\lambda_{i.}$.

2.3.2.4 CRFs Model Training

For all types of CRFs, the maximum-likelihood method can be applied for parameter estimation. That means, training the model is done by maximizing the log-likelihood $\mathcal{L}$ on the training data T:

$$\vec{\mathcal{L}}(T) = \sum_{(\vec{x},\vec{y}) \in T} \log P(\vec{y}|\vec{x})$$

In this case $(\vec{x}, \vec{y})$, is the sequence of observations with their respective sequence of tags taken from the training corpus.

$$= \sum_{(\vec{x},\vec{y}) \in T} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right)}{\sum_{\vec{y}' \in Y} exp\left(\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_{j}, \vec{x}, j\right)\right)} \right) \right] \quad (8).$$

To avoid over fitting the likelihood is penalized with the term,

$$-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

The parameter σ^2 models the trade-off between fitting exactly the observed feature frequencies and the squared norm of the weight vector and it is constant value.

For the derivation, the notation of the likelihood function ζ (T) is reorganized:

$$\vec{\mathcal{L}}(T) = \sum_{(\vec{x},\vec{y})\mathcal{E}\,T} \left[\log\left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i\, f_i(y_{j-1}, y_j, \vec{x}, j)\right)}{\sum_{\vec{y}'\,\epsilon y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i\, f_i\left(y'_{j-1}, y'_{j}, \vec{x}, j\right)\right)} \right) \right] - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2} \qquad (9).$$

$$= \underbrace{\sum_{(\vec{x},\vec{y})\mathcal{E}\,T} \sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i\, f_i(y_{j-1}, y_j, \vec{x}, j)}_{A} \underbrace{- \sum_{(\vec{x},\vec{y})\mathcal{E}\,T} \log \underbrace{\sum_{\vec{y}'\,\epsilon y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i\, f_i\left(y'_{j-1}, y'_{j}, \vec{x}, j\right)\right)}_{Z_{\vec{\lambda}}(\vec{x})}}_{B} \underbrace{- \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}}_{C} \qquad (10).$$

The partial derivations of $\mathcal{L}(T)$ by the weights λ_k are computed separately for the parts A, B, and C. The derivation for part A is given by:

$$\frac{\partial}{\partial \lambda_k} \sum_{(\vec{x},\vec{y})\mathcal{E}\,T} \sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i\, f_i(y_{j-1}, y_j, \vec{x}, j) = \sum_{(\vec{x},\vec{y})\mathcal{E}\,T} \sum_{j=1}^{n} f_k(y_{j-1}, y_j, \vec{x}, j) \qquad (11).$$

The derivation for part B which corresponds to the normalization is given by:

$$\frac{\partial}{\partial \lambda_k} \sum_{(\vec{x},\vec{y})\epsilon T} \log Z_{\vec{\lambda}}(\vec{x}) = \sum_{(\vec{x},\vec{y})\epsilon T} \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \frac{\partial Z_{\vec{\lambda}}(\vec{x})}{\partial \lambda_k}$$

$$= \sum_{(\vec{x},\vec{y})\epsilon T} \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \sum_{\vec{y}'\epsilon Y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right) . \sum_{j=1}^{n} f_k\left(y'_{j-1}, y'_j, \vec{x}, j\right)$$

$$= \sum_{(\vec{x},\vec{y})\epsilon T} \sum_{\vec{y}'\epsilon Y} \underbrace{\frac{1}{Z_{\vec{\lambda}}(\vec{x})} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right)}_{=P_{\vec{\lambda}}(\vec{y}|\vec{x})\ see\ equation\ (3)} . \sum_{j=1}^{n} f_k\left(y'_{j-1}, y'_j, \vec{x}, j\right)$$

$$= \sum_{(\vec{x},\vec{y})\hat{\epsilon}\, T} \sum_{\vec{y}'\,\epsilon y} P_{\vec{\lambda}}(\vec{y}'|\vec{x}) \sum_{j=1}^{n} f_k\left(y'_{j-1}, y'_j, \vec{x}, j\right) \quad (12)$$

Part C, the derivation of the penalty term, is given by:

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2}. \quad (13).$$

Equation 11, the derivation of part A, is the expected value under the empirical distribution of a feature f_i:

$$\tilde{E}(f_i) = \sum_{(\vec{x},\vec{y})\hat{\epsilon}\, T} \sum_{j=1}^{n} f_i(y_{j-1}, y_j, \vec{x}, j) \quad (14).$$

Accordingly, equation 12, the derivation of part B, is the expectation under the model distribution:

$$E(f_i) = \sum_{(\vec{x},\vec{y})\mathcal{E}\, T} \sum_{\vec{y}' \epsilon y} P_{\vec{\lambda}}(\vec{y}'|\vec{x}) \sum_{j=1}^{n} f_i \left(y'_{j-1}, y'_{j}, \vec{x}, j \right) \quad (15).$$

The partial derivations of $\mathcal{L}$ (T) can also be interpreted as:

$$\frac{\partial \mathcal{L}\,(T)}{\partial \lambda_k} = \tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2}. \quad (16).$$

To get the maximum by the approximation of the first derivation,

$$\tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} = 0$$

From this it is possible to calculate each weighting value of $\boldsymbol{\lambda_k}$ for each features f_k.

Computing $\tilde{\mathbf{E}}(\mathbf{f_i})$ is easily done by counting how often each feature occurs in the training data. Computing $\mathbf{E}(\mathbf{f_i})$ directly is impractical because of the high number of possible tag sequences $|Y|$. In a CRF, sequences of output variables lead to enormous combinatorial complexity. Thus, a dynamic programming approach is applied, known as the Forward-Backward algorithm which is beyond the scope of this study to explain [31].

2.4 Feature space for NER

Features are descriptors or characteristic attributes of words designed for algorithmic consumption. A boolean variable is an example of a feature which has the value true if a word is capitalized and false otherwise. Feature vector representation is an abstraction over text where

typically each word is represented by one or many boolean, numeric and nominal values. For example, a hypothetical NER system may represent each word of a text with 3 attributes. The first one is, a boolean attribute with the value true if the word is capitalized and false otherwise; second, a numeric attribute corresponding to the length in characters of the word; and lastly, a nominal attribute corresponding to the lowercased version of the word.

In this scenario, the sentence "The president of Apple eats an apple." excluding the punctuation, would be represented by the following feature vectors:

<true, 3, "the">, <false, 9, "president">, <false, 2, "of">, <true, 5, "apple">, <false, 4, "eats">, <false, 2, "an">, <false, 5, "apple">"

The next section describes word level and list lookup features that are used to recognize and classify named entities.

2.4.1 Word-level features

Word-level features are related to the character composition of words. They specifically describe word case, punctuation, numerical value and special characters. Table2.1 below lists subcategories of word-level features.

Table1 2.1: Word level features.

Features	Examples
Case	- Starts with a capital letter - Word is all uppercased -The word is mixed case (e.g., ProSys, eBay)
Punctuation	- Ends with period, has internal period (e.g., St., I.B.M.) - Internal apostrophe, hyphen or ampersand (e.g., O'Connor)
Digit	- Digit pattern - Cardinal and Ordinal - Roman number - Word with digits (e.g., W3C, 3M)
Character	- Possessive mark, first person pronoun - Greek letters
Morphology	- Prefix, suffix, singular version, stem - Common ending

Part-of-speech	- proper name, verb, noun, foreign word
Function	- Alpha, non-alpha, n-gram - lowercase, uppercase version - pattern, summarized pattern - token length, phrase length

*In Digit pattern, d*igits can contain important information like dates, percentages, and intervals etc. some patterns of digits for example in dates, two-digit and four-digit numbers can stand for years and when followed by an "s", they can stand for a decade; one and two digits may stand for a day or a month [23].

In Common word ending, morphological features are very important since they are essentially related to words affixes and roots. For instance, a system may learn that a human profession often ends in "ist" (journalist, cyclist) or that nationality and languages often ends in "ish" and "an" (Spanish, Danish, Romanian). Another example of common word ending is organization names that often end in "tech", and "soft" [23].

2.4.2 List Look up features

Lists are the privileged features in NER. The terms "gazetteer", "lexicon" and "dictionary" are often used interchangeably with the term "list". There are significant list lookup features like general list (eg. General dictionaries, capitalized nouns, common abbreviation), list of entities (eg. Organization; first names, last names; continent, country, state, city) and list of entity reminders (cues) (eg. Typical words in organization; Person title, name prefix, post-nominal letters; Location typical word, cardinal point) [23].

2.5 Amharic Language

In this section, the Amharic language with respect to part of speech, names and nouns in general will be discussed. It also discussed aspects of the language such as the letters of the language and its history.

2.5.1 Overview of Amharic

Amharic is one of the Semitic languages spoken in north central Ethiopia. Next to Arabic, it is the second most spoken Semitic language in the world and it is the official working language of the Federal Democratic Republic of Ethiopia. It is the second largest language in Ethiopia (after Afan Oromo, a Cushitic language) and possibly one of the five largest languages on the African continent. As a result it has official status and used nationwide. Despite it has large speaker population, the language has little computational linguistic resources [33].

The Amharic alphabet is called fidel, which grew out of the Ge'ez abugida-called in Ethiopian Semitic language. In modern written Amharic, each syllable pattern comes in seven different forms (called *orders*), reflecting the seven vowel sounds. The first order is the basic form; the other orders are derived from it by more or less regular modifications indicating the different vowels. The alphabet is written from left to right, in contrast to some other Semitic languages and consists of 33 consonants, giving 7*33=231 syllable patterns, or fidels [34]. In addition to the 231 characters, there are other non-standard alphabets which contain special features usually representing labialization. Refer to appendix-1 for a complete list of the symbols. Each alphabet represents a consonant together with its vowel. The vowels are fused to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order [35].

2.5.2 Grammatical Arrangement

In this section, the top level grammatical and morphological structure of the language will be covered. For categories of Amharic words, different researchers has classified into eight parts of speech and others in to five main parts of speech. The eight parts of speech are used in this research since POS tagger trained on WIC corpus uses this as a basic classification. These are

ስም (noun), **ግስ** (verb), **ቅፅል** (adjective), **ተውሳከ ግስ** (Adverb), **መስተዋድድ** (preposition), and **ተውላጠ ስም** (pronoun) and interjection categories[6, 37].

Verb: are words which can be placed at the end of a sentence. Some examples of Amharic verbs are **በላ**, **ሄዱ**, **ተናገረ.**

Adjective: any word that qualifies a noun or an adverb, which actually comes before a noun (e.g. **ጎበዝ ተማሪ**) and after an adverb (**በጣም ጎበዝ**).

Adverbs: are words that will be used to qualify a verb by adding extra idea on the sentence. The Amharic adverbs include **ትናንት**, **ገና**, **ዛሬ**, **ዶሎ**, and **እንደገና.**

Preposition: preposition is a word which can be placed before a noun and perform adverbial operations related to place, time, cause and so on; which can't accept any suffix or prefix; and which is never used to create a new word. It includes **ከ፣ ለ፣ ወደ፣ ስለ፣ እንደ**…

Pronoun: this category further can be divided as deictic specifier, which includes **ይህ**, **ያ**, **እሱ**, **እስዋ**, **እኔ**, **አንተ**, **አንቺ**…; quantitative specifier, which includes **አንድ**, **አንዳንድ**, **ብዙ**, **ጥቂት**, **በጣም**…; and possession specifier such as **የእኔ**, **የአንተ**, **የእሱ.**

Interjection: Like English, Amharic has many words or phrases used to express such emotions as sudden surprise, pleasure, annoyance and so on. Such Amharic words are called interjections. These Amharic interjections can stand-alone by themselves outside a sentence or can appear anywhere in a sentence.

Example:

ጎሽ!

ስራህ ጥሩ ነዉ ጎሽ!

2.5.3 Nouns

Like English, Amharic nouns are words used to name or identify any of a class of things, people, places or ideas or a particular one of these [33]. Nouns in Amharic are inflected by number, gender, case and definiteness. The inflection is achieved by either changing vowels or repeating consonants; and then adding the necessary affixes. For example, vowel ending nouns take the form **-ዎች** whereas consonant ending nouns add **-ኦች** at the end of their root word. The following examples illustrate such forms [37]:

በሬ + ዎች = በሬዎች

ተማሪ + ዎች =ተማሪዎች

ወንበር + ኦች = ወንበሮች

ቤት + ኦች = ቤቶች

In the case of some non-countable nouns repeats themselves in order to pluralize the quantity they are expressing for. The following examples illustrate such forms [37]:

ጥሬ ጥሬ-ኣ-ጥሬ [ጥራጥሬ]

ጌጥ ጌጥ-ኣ-ጌጥ [ጌጣጌጥ]

ትል ትል-ኣ-ትል [ትላትል]

Pronoun (**ተዉላጠ ስም**) and proper noun (**የተፀዉያ ስም**) are other categories of nouns. In Amharic, proper nouns (names) (**የተፀዉያ ስም**) are given to persons, locations, organizations, etc that specifically identify an entity in the real world [38]. To pluralize these nouns to indicate collection of similar names, we can use "**እነ-**". To illustrate this, see the next example [37]:

እሱ እነ እሱ [እነሱ]

አበበ [እነአበበ]

Let’s see some examples to understand the different names of the Amharic language,

የኢፌዴሪ ምክትል ጠቅላይ ሚኒስትር አቶ አዲሱ ለገሰ የቀይ መስቀል ማህበር የእድሜ ልክ አባል ሆኑ። አቶ አዲሱ የቀይ መስቀል ማህበር በሚያከናውናቸው ተግባራት ላይ ሁሉ በተቻላቸው አቅም እንደሚሳተፉ ገልጸው ሌሎችም የእሳቸውን አርአያ እንዲከተሉ ጥሪያቸውን አቅርበዋል። ....

የናሽናል አይል ኢትዮጵያ ኖክ ካመፓኒ ዋና ስራ አስፈጻሚ አቶ ታደሰ ጥላሁን በምረቃው ወቅት እንዳሉት ኩባንያው በኢትዮጵያ ነዳጅና ተዛማጅ ምርቶችን በማቅረብ የበኩሉን ድርሻ እየተወጣ ነው። .... .

የአፍሪካ ህብረት በሊቢያ ሲርት ባካሄደው 13ኛ መደበኛ ጉባኤ በልማት፣ በፀጥታና ደህንነት ጉዳዮች ላይ ነበር ትኩረት ያደረገው።

የምሥራቅ አፍሪካ የልማት በይነ መንግስታት ኢጋድ ከ3 ወራት በፊት በአዲስ አበባ ባካሄደው ጉባኤ የአልሸባብና የሂዝቡል ኢዝላም አቅም እየተጠናከረ መምጣቱን ገልጿል። አልቃይዳ ወደ ሶማሊያ ምድር መዝለቁን በግልጽ ማወጁ ደግሞ ሁኔታውን አስከፊ አድርጎታል።

ኤርትራ ከዚህ እኩይ ተግባሯ ካልታቀበች ደግሞ መቋዱሹ የአልቃይዳና የሌሎች ዓለም አቀፍ አሸባሪዎች የአፍሪካ መናኸሪያ ልትሆን እንደምትችል ስጋታቸውን የሚገልፁ አገራትና ተቋማት ቁጥር እየተበራከተ መጥቷል።

የተባበሩት መንግስታት ድርጅት በሶማሊያ ላሉ ጽንፈኞች ዋነኛ ድጋፍ የምታደርገው ኤርትራ መሆኗን በተደጋጋሚ አስታውቋል፤ ተከታታይ ማስጠንቀቂያዎችንም ሲሰጥ ቆይቷል።

የኤርትራ የማስታወቂያ ሚኒስትር አቶ አሊ አብዱ ለመገናኛ ብዙሀን የሰጡትን መግለጫ እንኳ ብንመከለከት የህብረቱ ውሳኔ አያስጨንቀንም ነው ያሉት።

ለቢቢሲ አምባሳደር አርአያ ደስታም ይህንኑ አቋም አራምደዋል።

«እኛ ከአሸባሪዎች ጋር ምንም አይነት ግንኙነት የለንም፤ ክሶቹ በአጠቃላይ መሰረተ ቢስ ናቸው፤ በቀጠናውም ሰላም እንዲሰፍን ነው እኛ ፍላጎታችን... » የሚሉ አባባሎች ከኢሳያስ አፈወርቂም ሆነ ከባለስልጣኖቻቸው ይሰማሉ።

As we can see from the above Amharic news which is collected from the Ethiopia’s Radio and Television Agency (ERTA) web site, the texts with red color are names of persons, with blue color are names of organization, and with violet color are names of locations.

The main objective of this thesis is to recognize those proper nouns and identify their categories. In named entity recognition system, part of speech tagging will also have immense advantages to extract the exact names.

2.5.3.1 Named Entities (NEs)

An entity is some object in the real world such as, a place, person or organization. A named entity is an entity that refers to named object which can be proper name, acronym, nickname or

abbreviation [4]. Some examples of named entities are: ኮካኮላ ቦትሊነግ ካመፓኒ፣ አዋሽ ባንክ፣ አዲስ አበባ and ሞላ መንገሻ.

Examples of the above names can be grouped into the categories: ORGANIZATION, LOCATION and PERSON names which are identified in the scope of this research.

Person names (PER)

People may be identified by name, nickname or alias. Names of deceased people, as well as fictional human characters appearing in movies, television, books and so on, should be taken as PERSON entities. Religious deities should also be taken as PERSON. In addition to this family names must also be taken as PERSON [4].

For example:

ወልደ ሃይለ - this is family name which indicates for all descendants of **ሃይለ**.

ታደሰ አህመድ -person name.

Most person names are preceded by titles, roles and honorifics such as "**አቶ**." and "**ፕሬዝደንት**". They serve as a feature to identify names of persons in texts.

[የአዲስ አባባ ንግድ ምክር ቤት] [**ምክትል ፕሬዝደንት**] [**ስለሞን ግዛዉ**] …
ORG Title PER

[የፍትህ] [**ሚንስትር**] [**አቶ**] [**በላይ በዛብህ**] ….
ORG Title Title PER

[አፈ ጉባኤ] [**ተሾመ ቶጋ**] …..
Title PER

የሲምፖዝሙ [**አሰተባባሪ**] [**አቶ**] [**ዳኜ**] **እንደገለጹት** . . . .
Role Title PER

Organization names (ORG)

Organization entities are limited to corporations, institutions, government agencies and other groups of people defined by an established organizational structure [4]. Let's see the following examples:

Business Organization: **አዋሽ ኢንሹራንስ ኩባንያ**

Multinational organizations: **የአፍሪካ ህብረት ድርጅት፣ የአዉሮፓ ህብረት ድርጅት**

Political parties: **ቅንጅት ላንድነትና ለዲሞክራሲ, ኢህአዴግ**

Sports teams: **ቅዱስ ጊወርጊስ ክለብ**

Military groups: **ታሚል ታይገር**

Many other kinds of entities refer to facilities or buildings that are primarily defined by their established organizational structure, and can do things like issue statements, make decisions, hire people, raise money and so on. A mention of such an entity should be considered as an ORGANIZATION when it functions like an ORG in the document [4]. These include things like:

Churches and other religious institutions:

[ኦርቶዶክስ ተዋህዶ ቤተክርስቲያን]

Hotels: **በቀለ ሞላ ሆቴሎች ድርጅት**

Museums: **የኢትዮጵያ ብሔራዊ ሙዚየም**

Universities: **አዲስ አበባ ዩኒቨርሲቲ**

Government office: **የጠቅላይ ሚኒስቴር ቢሮ**

Location names (LOC)

Location-related texts that are tagged as LOCATION include named continents, countries, provinces, cities, regions, districts, towns, villages, airports, highways, street names, factories, manufacturing plants, street addresses, oceans, seas, straits, channels, rivers, islands, lakes, national parks, mountains, fictional or imaginary locations [4].

Sometimes, there are situations where organization names can be location names and vice versa. In this case to differentiate whether an entity name is for location or organization, it is better to take in to consideration for the following definitions. ORG-refers to the organizational structure, and is acting like an agent (issuing a statement, making a decision, hiring people, raising money, etc.) and LOC- names are names that refers to the physical structure, rather than the people/groups who run it. Let's see some examples for this:

ዛሬ ማለዳ ላይ [ከፔንታጎን] የወጣዉ ዜና እንደሚያመለክተዉ ከሆነ... ::

in this case [**ፔንታጎን**] is name of organization.

ትናንት ምሽት [ፔንታጎን] ላይ በደረሰዉ ፍንዳታ 5 ሰዎችን ገደለ።።

[**ፔንታጎን**] is name of location.

Person names can also be location and organization names. Some examples are illustrated below:

[ሀይሌ ገብረ ስላሴ] ትናንት በዙሪክ በተካሄደዉ የ10000 ሜትር ሩጫ ሪከርድ በመስበር አሸነፉ::

In this case **[ሀይሌ ገብረ ስላሴ]** is name of a person.

በትናንትናዉ እለት ሀይሌ ገብረ ስላሴ ጎዳና ላይ ሁለት ሚኒባሶች ተጋጩ::

In this sentence **[ሀይሌ ገብረ ስላሴ]** is name of location.

Chapter Three

Related works

In this chapter, related works of different researchers in the area of named entity recognition were reviewed for different languages. To the best knowledge of the researcher, there is no research work done in Named Entity Recognition for Amharic Language. Under this chapter different papers on NER for different languages will be discussed. The chapter is organized into sections based on approaches, size and type of corpus used, the features used, and experimental result of the research.

3.1 Approaches

There has been considerable amount of work on NER in different languages such as English, Indian languages, Arabic, Chinese, etc. and much of the works are based on hand crafted rules, lookup form from the list (gazetteers), or data driven approaches exploiting the statistical properties of the language. But under this chapter more focus is given for papers that use statistical or hybrid approaches as it is the current state of the art.

The work in [9] reported about the development of NER system for South and South Eastern Indian languages, particularly for Bengali, Hindi, Telugu, Oriya and Urdu as part of the IJCNLP-08 NER Shared Task. In this research, statistical Conditional Random Fields (CRFs) model has been used to identify NEs. Similarly the work in [22], [12] and [11] used a conditional random field model for development of NER system for Hindi, Arabic and Chinese languages respectively. The research work in [12] is a comparative study of Arabic named entity recognition. Here the researchers conducted experiment to investigate the performance of the CRF model for Arabic NER task comparing the obtained result with their previous experiments which have been conducted using a maximum entropy approach.

The work in [2] was about the development of named entity recognition for Bengali. In comparison with the above three research papers, in this study the approach used was support vector machine which performs classification by constructing an N dimensional hyper plane that optimally separates data into categories. In this study, both the training and classification processes were carried out by YamCha toolkit which is an SVM based tool for detecting classes in documents and formulating the NER task as a sequential labeling problem.

The other approach is HMM from which a named entity recognition (NER) system is built to recognize and classify names. Different researchers have used this approach like the work in [10]. In this work, the researcher has justified that HMM is better than maximum entropy, decision tree and some other approaches. Among those approaches, the evaluation performance of HMM is higher than those listed approaches. This is due to its better ability of capturing the locality of phenomena, which indicates names in text. Moreover, the researchers mentioned that, HMM seems more and more used in NE recognition because of the efficiency of viterbi algorithm used in decoding the NE-class state sequence. But in this research, the researchers didn't incorporate the CRFs approach from the list of approaches that they have mentioned as a comparison. As it was mentioned in section 2.3.2.3, CRFs is a better approach than HMM.

The work in [9] presents a hybrid approach for NER for English. In this paper, the researchers combined three different approaches taking in to consideration the pros and cons of each approach and then the gap of one approach is filled by the others to get the best performance of the system in order to be practically usable. In this study, ME, HMM and hand crafted grammatical rules are combined and applied to this system. The NER task demonstrates characteristics that can be exploited by all three techniques. For example, time and monetary expressions are fairly predictable and hence processed most efficiently with handcrafted grammar rules. Name, location and organization entities are highly variable and thus provide themselves to statistical training algorithms such as HMMs. And then HMM generates the standard MUC tags (person, location and organization). Finally, many conflicting pieces of information regarding the class of a tag are frequently present. For this, a ME approach works well in utilizing diverse sources of information in determining the NE sub-categorization such as city, airport, government, etc. from the basic tags).

3.2 Features used

Experiments were carried out to find out most suitable features for NE tagging task. The main features for the NER task have been identified based on different possible combination of available word and tag context.

In [22], it was considered language independent features as well as language dependent features. The language independent features include the contextual words, prefix, and suffix information of all the words in the training corpus, several digit features and the frequency features of the

words particularly linguistic features are considered by the system for Bengali and Hindi. Linguistic features of Bengali include the set of known suffixes that may appear with named entities, clue words that help in predicting the location and organization names, words that help to recognize measurement expressions, designation words that help in identifying person names, the various gazetteer lists like the first names, middle names, last names, location names and organization names. As part of linguistic features for Hindi, the system uses only the lists of first names, middle names and last names along with the list of words that helps to recognize measurements. No linguistic features have been considered for Telugu, Oriya and Urdu. In the study it was mentioned that the use of linguistic features improves the performance of the system. In a similar fashion the work in [2] put the features that were used to recognize names in general format.

F={ W_{i-m},…, W_{i-1}, W_i, W_{i+1},…, W_{i+n}, |prefix|≤n, |suffix|≤n, previous NE tags, POS tags, First Word, Digit information, Gazetteer lists}.

Here the researchers have used different combination from the above set for inspecting the best feature set for NER task.

Some researchers also classify those features in different ways. In the work [10], the system is able to apply and integrate four types of internal and external features. (1) Simple deterministic internal feature of the words, such as capitalization; (2) internal semantic feature of important triggers, such as organization suffix (Ltd), time suffix (a.m), etc.; (3) internal gazetteer feature; and (4) external macro context feature.

Moreover, in [8], the researchers investigated the impact of using different sets of features in two discriminative machine learning frameworks, namely, SVMs and CRFs using Arabic data. According to the researchers, different classes are sensitive to different features. Hence, the researchers discovered the optimum feature set per NE class since features that are discriminative for one NE class might not be for another class. In the process, it is decided on an optimal set of features for each NE class. Finally, the different classifiers are combined to create a global NER system.

As a summary, most of NER systems for different languages may use combination of the following features:

- Word context feature: Previous and next words of a particular word might be used as a feature. Here different window size can be used.
- Part of speech can also be used as a feature in different languages.
- Word suffix: Word suffix information is helpful to identify NEs. A fixed length word suffix of the current and surrounding words might be treated as feature.
- Word prefix: Prefix information of a word is also helpful. A fixed length prefix of the current and the surrounding words might be treated as features.
- Named Entity Information: The NE tag of the previous word can also be considered as the feature, i.e., the combination of the current and the previous output token can be considered.
- Gazetteer list: The simplest approach of using these gazetteers is to compare the current word with the lists and make decisions. But this approach is not good, as it can't resolve ambiguity. So, it is better to use these lists as the features of the CRF. If the current token is in a particular list, then the corresponding feature is set to 1 for the current/previous/next token otherwise, set to 0.

3.3 Data used for training and testing

In different researches for NER that are reviewed, different types of corpuses were used for training and testing. For example, the work in [22], the training data were provided for five different Indian languages, namely Bengali (122,467 tokens), Hindi (502,974 tokens), Telugu (64,026 tokens), Oriya (93,173 tokens) and Urdu (35,447 tokens) in Shakti Standard Format. In all the languages, the training data were labeled with twelve NE tags as defined for the IJCNLP-08 NER shared task tag set[1].

On the other hand, the data used in the work of [29] on Hindi language, for training of the model was taken from tourism domain and is collected from web. The data was labeled with the Hindi POS tagger and the named entities were tagged manually. During the experiment, the researchers used 150 sentences for testing and 3000 sentences for training. In addition, the study in [2], for

[1] http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=3

training and testing set, they have used a partially NE tagged Bengali news corpus developed from the archive of a widely read Bengali newspaper. The corpus contains around 34 million word forms in ISCII (Indian Script Code for Information Interchange) and UTF-8 format. Out of 34 million word forms, a set of 150K word forms has been manually annotated. Around 20K NE tagged corpus is selected as the development set and the rest 130K word forms are used as the training set of the SVM based NER system.

In addition to the above studies, the work in [9], has used data from MUC-6 and MUC-7 NE tasks. The system was used 1330KB from MUC-6 and 708KB from MUC-7 NE tasks for training and also for testing, 121KB from MUC-6 and 186KB from MUC-7 respectively. And the work in [8], have tested the system on MUC-7 dry run data; this data consists of 22,000 words and represents articles from The New York Times.

Finally, NER system for Arabic language in the work [12] and [8] used different source of data. The work of [12], the researchers used ANERcorp to train and test the CRF model. It is composed of a training corpus and a testing corpus annotated especially for the NER task. It was collected from both news wire and other web resources. The ANERcorp contains more than 150,000 tokens (11% of the tokens are part of NE). This corpus was used in both systems i.e the previous system which was developed using maximum entropy approach and also in the second version which used the CRFs approach. But in [8], the researchers used the standard sets of ACE 2003, ACE 2004 and ACE 2005 data sets for training, development and testing. All the data sets comprise broadcast news and newswire genres. ACE 2004 includes an additional news wire data set from the Arabic TreeBank (ATB). ACE 2005 includes a different genre of weblogs. The ACE 2003 data defines four different NE classes: Person (e.g. Albert Einstein), Geographical and Political Entities (GPE) (e.g. Kazakhstan), Organization (e.g. Google Co.) and Facility (e.g. the White House). Whereas in ACE 2004 and 2005, two NE classes are added to the ACE 2003 tag-set: Vehicles (e.g. Rotterdam Ship) and Weapons (e.g. Kalashnikov).

3.4 Experimental results

Performance of all reviewed papers is reported using, precision, recall and F-measure. Here precision (P) measures the number of correct NEs in the answer file (Machine tagged data) over the total number of NEs in the answer file and recall (R) measures the number of correct NEs in the answer file over the total number of NEs in the key file (gold standard). F-measure (F) is the

harmonic mean of precision and recall: $F = \frac{(\beta^2+1)PR}{\beta^2 R+P}$ when β^2=1 [26]. Based on this, the work in [22] has shown that the CRF based NER system performs best for Bengali with maximal F-measure of 55.36%, nested F-measure of 61.46% and lexical F-measure 59.39%. Next to Bengali, CRF based NER for Hindi has demonstrated the F-measures of 35.37%, 36.75% and 33.12%, respectively for maximal, nested and lexical matches.

In the work of [29], the test was conducted in three different ways i.e. the first case was conducted without using POS tagger but using context features. In the second case, NE tags were included which helps to disambiguate some confusing classifications like in case of organization names; Lastly, it was used POS tagger as a result there was better improvement than the others. The results shown in the three cases as mentioned in the three tables (p-107 of [22]), accuracy in case of organization, whatever combination it was taken, is quite low compared to other NEs. This is due to the fact that most of the names of organizations are multiword and even in some cases comprise of person name, location name too. In addition, experimental results of the test in the work of [2] showed that the overall average Recall, Precision and F-Score of 94.3%, 89.4% and 91.8%, respectively.

Experimental result for the work of [10] has shown that on MUC-6 and MUC-7 English NE tasks achieved F-measure of 96.6% and 94.1% respectively. Moreover, the work in [9] has shown that the scoring program computes both the precision and recall for each category, and combines these two measures into f-measure of 93.39% as the weighted harmonic mean.

Finally, Experimental result for Arabic NER in the work of [12] , showed that the CRF approach gave a better performance in comparison with the previously applied approaches for ANERsys i.e they have found precision, recall, and F-measure of 86.90%, 72.77%, and 79.21% respectively after combining all the features they have used. But in the previous version of ANERsys 2.0 which used maximum entropy approach, the result showed 70.24%, 62.08%, 65.91%, for precision, recall and F-measure respectively. In addition, in the work of [8], the researchers reported that F-score of 83.5% for β=1 as best result for the ACE 2003 broadcast news data.

3.5 Conclusion

Based on the above studies, CRF model is the current state of the art for named entity recognition system since most of the research papers reviewed, use CRF model. With regard to features, the most commonly used features are contextual features, POS tags, suffix and prefix, named entity tag information and gazetteer lists. Moreover, different studies use different data type and size but what make them similar is, the more the training data used the better the performance will be.

As a summary table 3.1 shows type of language, approach, tool, corpus used, features used and performance of all papers reviewed.

Table 3.1 NER experimental results from literature.

Language	Approach	Tool	Corpus name	Corpus size	Features used	Performance (%)			Remark
						P	R	F	
South East Asian languages (Bengali)	CRF	C++ based OpenNLP CRF++ package		122,467 tokens for training, 30,505 for testing	contextual words, prefix, and suffix information of all the words, previous NE tag, POS tags, gazetteer list	51.63	59.60	55.36	in this study it was considered 5 languages but I have mentioned Bengali that has the highest performance(maximal)
Hindi	CRF			150 sentences for testing and 3000 sentences for training	some specific suffixes, context feature, context wordlist(clues), POS of words,				
Bengali	SVM	YamCha toolkit	Bengali news corpus	150K words used for training and 130K words used for testing out of 34 million words	Contextual words, prefix, and suffix information of all the words, previous NE tag, POS tags, gazetteer list.	94.3	89.4	91.8	
					Simple deterministic internal feature of the words, internal semantic feature of important	95	95.7	95.35	Average F-measure, P and R of MUC6 and MUC 7

English	HMM		MUC-6 and MUC-7	2038KB for training and 277KB for testing	triggers, such as organization suffix (Ltd), time suffix (a.m), etc., internal gazetteer feature, and external macro context feature				
>>	Hybrid		MUC-7 dry run data	22,000 words	Ruled based for numerals, and dates,			93.39	
Arabic	CRF		ANERcorp	150,000 tokens	word and the context in which it appeared in the text	86.9	72.77	79.21	11% of the corpus is NEs
>>	CRF and SVM		ACE2003 data sets	12.41K for training 5.63K for testing	Contextual, lexical, and gazetteers, POS, nationality, morphological features.			83.5	
Chinese	CRF		CityU	80% and 20% of the corpus for training and testing respectively	text feature, POS feature, and small-vocabulary-character lists feature	92.76	81.81	86.94	

Chapter Four

Design and implementation of the system

Based on the description of the previous chapters, NER involves the identification and classification of named entities such as person names, location names, and organization names. Moreover, in the taxonomy of computational linguistics, NER falls within the category of information extraction which deals with the extraction of specific information from given documents.

In this chapter, the design and implementation of CRF based NER system for Amharic is discussed. This chapter is organized as follows: the first section talks about corpus used for training and testing for both POS tagger and NER. Section 4.2 describes over view of the architecture; section 4.3 describes about model generation; section 4.4 describes about the classifier; and finally section 4.5 describes about performance evaluation.

4.1 Corpus description

Mostly, NLP that uses statistical approach needs huge amount of data. The success or failure of most NLP applications depends on the quality and availability of appropriate data. The data used in computational linguistic tasks generally take the form of corpora. Corpora can be divided into two categories: annotated corpora and unannotated corpora. Unannotated corpora are simply large collection of raw text, where as annotated corpora add additional information to the text, such as part-of-speech tags and named entity tags. Annotated corpora with appropriate part of speech and named entity tags are useful to train part of speech taggers and named entity recognizer respectively.

In this work, part of the WIC Amharic corpus which is prepared by the Ethiopian Languages Research Center of Addis Ababa University in a project called "The Annotation of Amharic News Documents" has been used. The project was meant to tag manually each Amharic word in its context with the most appropriate parts-of-speech. This corpus is prepared in two forms i.e. in Amharic version (using Ge'ez fidel) and in transliterated format using Latin characters. Since I couldn't get the Amharic version, I have used the transliterated corpus. As shown in Table 4.1, the corpus has 210,000 words collected from 1065 Amharic news documents of Walta Information Center, a private news and information service located in Addis Ababa, Ethiopia.

Table 4.1 WIC corpus statistics

No. of Documents	No. of Sentences	No. of Words	No. of tag sets
1065 Amharic news documents	7003	210,000	30

The WIC corpus has a number of problems such as tag set errors. The following tags has been identified as an error since it is not in the list of tag sets: AADJP, AD, ADG, ADJVC, ADPC, ADR, AUX, CN, CONJC, INT, J, JPC, M, N;P, NDJ, NPNP, NPP, NU, NUNCR, NUNP, NUOR, Np, NumcRn, PC, PNC, PRINP, PROON, PROPN, PROPP, PUC, PUNC`, PUVC, UN, UNC, VNV, VO, VPUNC, VREP, VREV, Vp, body, fidel, sera, title. It also contains the following string repeatedly which is not part of the corpus:

"//sera /copyright copyright1998-2002WaltaInformationCenter//copyright //body //document -/document /filename mes07a1.htm//filename -/title /fidel //fidel /sera . . . //sera //title /datelineplace="adisabeba"month="meskerem"date="7/1994/(WIC)/"/ -/body /fidel 1520//fidel /sera"

Due to those problems, it is very difficult to take the whole corpus as it is to train POS tagger for Amharic.

4.1.1 Corpus preparation for POS tagger

POS tagging is the process of assigning a POS or other lexical class marker to each word in a corpus. POS tagger is a tagging system which assigns a tag for each word in a sentence automatically. POS tagging, not only assign a word to the accurate part-of-speech tag, but also provides other relevant information such as the inflectional categories of the classes, for example, number, person, location, and organization. Due to this reason, Part-of-speech of words is used as a feature in Amharic NERS. As a result, POS tagger is trained using Ling Pipe tool and this tagger is used in ANER system.

To train the POS tagger, randomly selected sentences from WIC corpus has been taken. Each sentences started in a new line. Using the LingPipe tool, the system was trained on a total of

14363 tokens as mentioned in Table 4.2 and finally generates the HMM model. And finally tested on 3936 tokens.

Table 4.2 Corpus statistics for POS tagger

No. of tokens for training	14363
No. of tokens for testing	3936
Total No. of sentence	633(9.04% of the total sentences of WIC)
Total No. of tokens	18299

4.1.2 The POS tags used in the WIC corpus

The tag sets identified as basic POS are nouns (N), pronouns (PRON), adjectives (ADJ), adverbs (ADV), verbs (V), prepositions (PREP), conjunctions (CONJ), and interjection (INT). Since punctuations should also be annotated, we have included a PUNC tag in the tag set. To give a room for tagging difficult or problematic words that the annotators may face, they have included a UNC tag (unclassified). The ten basic classes were then further divided into a total of thirty subclasses as shown in table 4.3. Refer detail definition of sub classes in [42].

Table 4.3 POS Tag sets for WIC corpus

No.	Basic Class	Code of the tag
1.	Noun	VN NP NC NPC N
2.	Pronoun	PRONP PRONC PRONPC PRON

3.	Verb	AUX VREL VP VC VPC V
4.	Adjective	ADJP ADJC ADJPC ADJ
5.	Numeral	NUMCR NUMOR NUMP NUMC NUMPC
6.	Preposition	PREP
7.	Conjunction	CONJ
8.	Adverb	ADV
9.	Punctuation	PUNC
10.	Unclassified	UNC

4.1.3 Corpus preparation for ANER system

The system needs three types of files such as training data, a development data, and a test data. The learning methods were trained with the training data. The development data is used for tuning the parameters of the learning methods. After completing the training and generating the model, the system will be tested on testing data to evaluate the performance of the system. The data of those three files have been taken from WIC corpus which is 4.95% of the total as most of the sentences do not have named entities in it.

Sentences that contain at least one named entity have been taken from WIC corpus. POS of words were removed and tagged with named entity tag sets with IOB notation. The data used

was already subjected to some linguistic preprocessing like tokenization and manual named entity tagging of training data, development, and test data following CoNLL2002 format. This format keeps all the sentences of the three files separated by an empty line. Every word in a sentence will be kept on one line together with the NE tag separated by a space. The following shows sample format taken from the training data file:

lElaw O
yekbr O
Ingdana O
asteyayet O
seCi O
besemEn B-ORG
xewa I-ORG
zon I-ORG
yew`ha B-ORG
ma`Idnna I-ORG
Inerji I-ORG
memriya I-ORG
`halafi O
ato O
kebede B-PER
gerba I-PER
bebekulacew O
beIyandandu O
guday O
lay O
yeguba`Ew O
tesatafiwoc O
yeme`selacewn O
hesab O
bene`Sanet O
yeseTubetna O
kalefew O

shtet O
lememar O
yalacew O
zgjunet O
drjtun O
yemiyaTenakrew O
new O
blewal O
:: O

mktl O
teqlaymini`strna O
mekelakeya O
mini`str O
tefera B-PER
walwa I-PER
leamErikaw O
yemekelakeya O
mini`str O
lekbur O
donaldramsfild B-PER
belakut O
debdabE O
beniwyorkna B-LOC
waxngten B-LOC
betekaHEdut O
Cfn O
yeaxebariwoc O
Tqat O
yederesewn O
gudat O
kelb O

Indasazenacew O

gel`Sew O

drgitun O

awgzewal O

:: O

Words tagged with O (Other) are words which are not named entities. The B-XXX tag is used for the first word in a named entity of type XXX and I-XXX is used for all other words in named entities of type XXX. The data contains entities of three types: persons (PER), organizations (ORG), and locations (LOC). The tagging scheme is a variant of the IOB scheme originally put forward by Ramshaw and Marcus [39].

4.2 Architecture overview

As it was mentioned in section 1.5, three major named entity categories were considered: person, location and organization. Under this chapter, the architecture (design) of the system shown in Figure 4.1 will be presented. The diagram shows the overall functionality of the Amharic named entity recognition system.

In the architecture, there are three major phases preprocessing, training and testing: The preprocessing phase contains two modules parser and tokenizer. The parser reads data from training and development data and also checks the format of the training and development corpus as the corpus is in CONLL 2002 format and finally it gives the tokens and tags to the next phase. The tokenizer takes the input text supplied from a user and tokenizes it into a sequence of tokens. It also puts a line between sentences as an indication for end of a sentence and finally gives the tokenized sentences to testing phase.

The second phase is the training phase which contains the Feature-Extractor, chunker and training of ANER with CRFs. The training phase performs parameter estimation and generates CRFs model. The third phase is testing phase which performs the recognition of the named entities using the model which is generated by the training phase. In general, those three phases are grouped in to model generation and classifier.

4.3 Model generation: is a process of generating the ANER model using the validation and training data. This phase involves preprocessing and training phase.

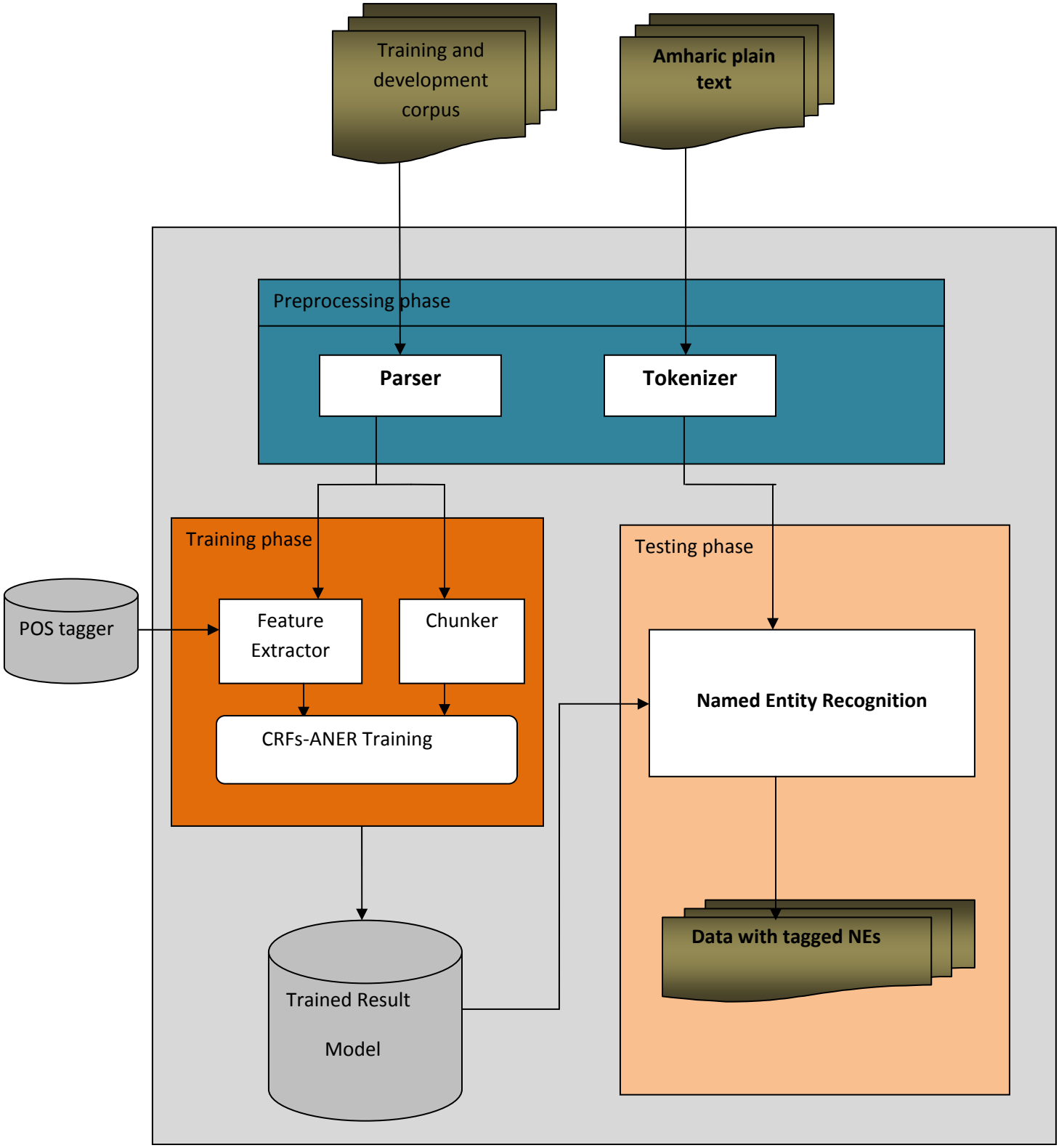

Figure 4.1 Architecture of Amharic NER system.

4.3.1 Preprocessing phase

This phase contains "Parser" and "Tokenizer" modules. The parser reads data from training and development corpus , checks the legal token-tag sequence and finally gives the tokens and tags to the training phase. The legal tag sequences are described in Table 4.4. In the table, the first column lists tags schematically and the second column shows the legal tags that may follow them.

Table 4.4 legal tag sequence.

Tag	**Legal Following Tags**
O	O, B-*X*
B-X	O, I-*X*, B-*Y*
I-X	O, I-X, B-Y

Where, O stands for Other which is assigned for words that are not named entities; B shows beginning of named entity; I shows inner named entity; X and Y shows different named entity types. Note that the begin and in-tags have the same legal followers. Attempts to encode taggings with illegal tag sequences will result an error.

The parser also contains two regular expressions such as ignore and end of sentence (eos) regular expressions. If the ignore regular expression is matched, an input line is ignored. This is useful for ignoring empty lines and comments in some inputs. The "eos" regular expression recognizes lines that are ends of sentences. Whenever such a line is found, the zone currently being processed is sent to the next handler.

The tokenizer module reads the Amharic plain text from file and performs tokenization which is the process of breaking up a string into tokens. In Amharic, tokenization process is easier since every word is separated by a space. In addition end of a sentence can be identified using two consecutive colon characters (::). After tokenization every token is given to the testing phase.

4.3.2 Training phase

This phase has “Feature_Extractor”, “chunker” and the CRFs-ANER training process. “Feature-Extractor” will take the tokens and tags coming from the preprocessing phase. Moreover, “Feature-Extractor” will use POS tagger to know the POS of the tokens since it is used as a feature.

In order to efficiently compute the best tagging(s) given an input sequence, the features are restricted to depend only on local features of the output tags. In first-order chain CRFs, features may only depend on pairs of output tags, it means providing one tag of context in predicting the next tag. Suppose we have an input sequence of length N,(i.e. x[1],...,x[N]) and K output categories coded as integers 1,...,K. We extract features for each position based on the position in the input (1,...,N) and for each previous tag (from 1,...,K). For efficiency, it is factored the feature extractor into two parts, one of which pays attention to the previous tag, and one of which doesn't. In general, the “Feature_Extractor” has two modules. One which extracts node features and the other module extracts edge features. And finally converts nodes and edges to feature maps, which are then converted to vectors for use in CRFs.

Node features include POS, suffix, prefix, named entities categories of the current, previous and next token, suffix and prefix features with maximum length of four. And edge feature includes previous tags of tokens. After extracting those features it will send to the next process i.e. training process. Here is the algorithm of the node and edge features extractor:

The following are feature representations:

“BOS”, “EOS”, “Token”,”Token_Prev”, “Token_next”, “POS_Tag”, “POS_Tag_Prev”, “POS_Tag_Next”, “Token_Suffix”, “Previous_Token_Suffix”, “Next_Token_Suffix”, “Token_Prefix”,”Previous_Token_Prefix”, “Next_Token_Prefix”,

These are node feature representations with binary value of 0 by default. When those features feat with the coming tokens, there value will be set in to 1. Pseudo code for node feature extraction and edge feature extraction is shown in Figure 4.2 and Figure 4.3 respectively.

```
Node_feature_Extractor(int n)/// where n is the position of the token{
if token(n) is beginning of a sentence
    set feature key value of "BOS" to 1
if token(n) is end of a sentence
    set feature key value of "EOS" to 1
if token(n) is a token
    Concatenate feature key "Token" with token(n) and set its value to 1
if token(n-1) is a token and not beginning of a sentence
    Concatenate feature key "Token_Prev" with token(n-1) and set its value to 1
if token(n+1) is a token and not end of a sentence
    Concatenate feature key "Token_Next" with token(n+1) and set its value to 1
  // in this case the part of speech tagger is used to tag the tokens
if Tokentag(n) is a part of speech tag
     Concatenate feature key "POS_Tag" with tokentag(n) and set its value to 1
if tokentag(n-1) is part of speech tag  and not beginning of a sentence
         Concatenate feature key "POS_Tag_Prev" with tokentag(n-1) and set its value to 1
if tokentag(n+1) is part of speech tag  and not end of a sentence
          Concatenate feature key "POS_Tag_Next" with tokentag(n+1) and set its value to 1
if  prefixOfToken(n) has a prefix with maximum length of 4
         Concatenate feature key "Token_Prefix" with prefixOfToken(n) and set its value to 1
```

```
if  prefixOfToken(n-1) has a prefix with maximum length of 4 and not beginning of a sentence
      Concatenate feature key “Previous_Token_Prefix” with prefixOfToken(n-1) and set its value to 1
if  prefixOfToken(n+1) has a prefix with maximum length of 4 and not end of a sentence
      Concatenate feature key “Next_Token_Prefix” with prefixOfToken(n+1) and set its value to 1
if  SuffixOfToken(n) has a suffix with maximum length of 4
      Concatenate feature key “Token_Suffix” with SuffixOfToken(n) and set its value to 1
if  SuffixOfToken(n-1) has a suffix with maximum length of 4 and not beginning of a sentence
     Concatenate feature key “Previous_Token_Suffix” with SuffixOfToken(n-1) and set its value to 1
if  SuffixOfToken(n+1) has a suffix with maximum length of 4 and not end of a sentence
    Concatenate feature key “Next_Token_Suffix” with SuffixOfToken(n+1) and set its value to 1
Finally return these features
```

Figure 4.2 Node Feature extractor algorithm.

```
The edge features are represented by the key “Previous_tag”:
Edge_Feature_Extractor (int n, int k) //where n is position of the token and K is the category.
  If the position of the token is not at the beginning of  the sentence
    Concatenate the “Previous_Tag” key  with the tag(K) and set its value as 1
    Concatenate “Previous_Tag_Token_Category” with tag(k) +tokencat(n-1) and set its value as 1
    Finally return the features
```

Figure 4.3 Edge Feature Extractor

The other module in the training phase is the chunker which converts the tagging in to chunking i.e. if we have sequence of tokens like

Moges B-PER

Ahmed I-PER

The chunker will convert those tagged tokens into chunk set ((Moges Ahmed) PER) and the training process takes chunk set as an input.

Figure 4.4 shows the algorithm for the chunker:

```
Chunker(tokens, tags){
        String Chunkset="";// null string
        If the tag begins with "B-" then
                chunkset=(chunkset+' '+ Token) //concatenation
        while(next tokens_tag begins with "I-")
                chunkset=(chunkset+' '+Token);//concatenation
            return chunkset;
```

Figure 4.4 algorithm for chunker

The training process needs many more parameters in addition to the above inputs. A regressionPrior instance is one of the parameter which represents a prior distribution on parameters for linear or logistic regression. It has methods to return the log probabilities of input parameters and compute the gradient of the log probability for estimation. Secondly, annealing Schedule - Schedule for annealing the learning rate during gradient descent. Thirdly, minimum and maximum number of epochs for which to run gradient descent estimation. Finally reporter instance reports to which results are written, or null for no reporting of intermediate results during the training.

Taking those inputs, the CRF ANER training process performs parameter estimation such as λ_i for each feature function f_i based on the extracted features. Finally, generates the ANER CRFs model. The algorithm for CRFs is described in literature review section 2.3.2.

4.4 Classifier

Tokenizer and testing phase can be considered as a classifier. The Tokenizer phase in model generation is used in similar fashion in classifier. In testing phase, using the knowledge stored CRF model, it will detect and classify the named entities. Let's see the next section for detail explanation.

4.4.1 Inference

Given a chain-structured CRFs model, the general inference task is to find the label sequence that maximizes the joint conditional probability, which can be calculated efficiently through the Viterbi algorithm. Let Y be the set of possible labels, where | Y | =m. A set of m x m matrices $\{M_i (X) \mid i=0,\ldots, n-1\}$ is defined over each pair of labels y', y ε Y

$$M_i(y', y \mid X) = \exp\left(\sum_j \lambda_j f_j(y', y, X, i)\right).$$

By augmenting two special nodes y_{-1} and y_n before and after the sequence with labels **start** and **end** respectively, the sequence probability is

$$p(Y|X,\lambda) = \frac{1}{Z(X)} \prod_{i=0}^{n} M_i(y_{i-1}, y_i \mid X)$$

Z(X) can be computed from the M_i's but not needed in evaluation. Therefore, we only need to find the label sequence Y that maximizes the product of the corresponding elements of these n + 1 matrices. The Viterbi algorithm is the standard method that computes the most likely label sequence given the observation. It *grows* the optimal label sequence gradually by scanning the matrices from position 0 to n. At step i, it records all the optimal sequences ending at a label y,

∀ y ε Y (denoted by $Y_i^*(y)$), and also the corresponding product $P_i(y)$. The recursive function of this dynamic programming algorithm is as follows,

1. $P_0(y)=M_0(start, y \mid X)$ and $Y_0^*(y)=y$
2. For $1 \leq i \geq n$,
   $Y_i^*(y) = Y_{i-1}^*(\hat{y}).(y)$ and
   $P_i(y) = max_{y' \varepsilon Y} P_{i-1}(y') M(y', y \mid X)$
   Where $\hat{y} = argmax_{y' \varepsilon Y} P_{i-1}(y') M(y', y \mid X)$ and "." is the concatenation operator.

Figure 4.5 An algorithm for dynamic programming [41].

The optimal sequence is therefore $Y_{n-1}^* = [Y_n^*]_{0\ldots n-1,}$ which is the best path to the end symbol but taking only position 0 to position n-1[41].

4.5 Performance evaluation

CRF based ANER system was evaluated with respect to the training corpus which contains 8005 tokens. The evaluation was performed by comparing the system output to the human-annotated corpus in terms of the precision (P), recall (R) and their harmonic mean, the F-measure (F).

$$\boldsymbol{P} = \frac{\mathbf{T_P}}{(\mathbf{T_P} + \mathbf{F_P})}$$

And,

$$\mathbf{R} = \frac{\mathbf{T_P}}{(\mathbf{T_P} + \mathbf{F_N})}$$

Where, True Positives (T_P) are chunks produced by the CRF that match the reference chunking and False Positives(F_P) are chunks returned by the CRF that are not in the reference chunking. And lastly, False Negatives (F_N) are chunks in the reference chunking missed by the CRF.

These two measures of performance combine to form one measure of performance, the F-measure, which is computed by the uniformly weighted harmonic mean of precision and recall:

$$\mathbf{F} = \frac{\mathbf{RP}}{\mathbf{1/2(R + P)}}$$

Chapter Five

Experimental Result

In chapter four, the design of Amharic Named Entity Recognition system with its implementation are discussed. This chapter presents the process of the experiments conducted. Experiments are done in different scenarios so that the results are displayed accordingly with their explanations.

5.1 Named Entity Features

Feature selection plays a crucial role in CRF framework. Experiments were carried out to find out most suitable features for NE tagging task. The main features for the NER task have been identified based on the different possible combination of available word and tag context. The features also include prefix and suffix for all words. The term prefix/suffix is a sequence of first/last few characters of a word, which may not be a linguistically meaningful prefix/suffix. In this study, it has been considered different combination from the following set for inspecting the best feature set for the Amharic NER task:

F={ W_{i-m},…, W_{i-1}, W_i, W_{i+1},…, W_{i+n}, |prefix|≤n, |suffix|≤n, NE tags, POS tags, BOS, EOS, }.

In this case NE tags includes the previous and current NE tags of the tokens. In a similar fashion POS tags includes previous, current, and next POS tags of tokens.

Following is the details of the set of features that were applied to the Amharic NER task:

- Context word feature: Previous and next words of a particular word are used as a feature. We have considered the word window of size three, i.e., previous and next words from the current word for this language.
- Word suffix: Word suffix information is helpful to identify NEs. A fixed length word suffix of the current and surrounding words are treated as feature. In this work, it was made different experiments to determine the appropriate suffixes length and length up to five of the current word have been taken for this language since it gave better performance than using length up to 3, 4, and 6.

- Word prefix: A fixed length prefix of the current and the surrounding words might be treated as features. Here, the prefixes of length up to 4 have been considered.
- Beginning of a sentence (BOS): If the current token is the first word of a sentence, then this feature is set to 1. Otherwise, it is set to 0.
- End of a sentence (EOS): If the current token is the last word of a sentence, then this feature is set to 1. Otherwise, it is set to 0.
- Named Entity Information: The NE tag of the previous word is also considered as the feature, i.e., the combination of the current and the previous output token has been considered
- Part of speech - part of speech of tokens helps to identify nouns. Since all named entities are nouns, then POS of tokens is used as a feature to recognize named entities.

5.2 Part of speech tagger for Amharic

Since there is no POS tagger for Amharic which suits for this study, Amharic POS tagger was trained using a lingPipe tool. To train this system, 14363 tokens were used and tested on 3936 tokens. The performance of this POS tagger is 81.555% (total accuracy).

5.3 Experiment

In this study, a total of 10,405 tokens were taken from WIC and manually annotated with tags B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, and O (Other which is not a named entity). From the total tokens annotated, 10.1 % are named entities. This corpus is again divided for training, development and testing as shown in Table 5.1.

The CRF based Amharic NER system has been trained and tested on training and testing data. The training, development and test data statistics are presented in Table 5.1.

Table 5.1 Corpus statistics

Number of tokens in the training set	Number of tokens in the development set	Number of tokens in testing set	Total number of tokens
8005 tokens	1439 tokens	961 tokens	10405

To do the experiment, four different scenarios were considered. In the first scenario, all the features mentioned above were used and gave the performance as shown in Table 5.2. This scenario is taken as a baseline for the remaining experiments.

Table5.2. Performance of the ANER system with all the features (scenario one)

Recall	75%
Precision	72%
F-measure	73.47%

In the second scenario, all the features except part of speech tags were used. In this case part of speech tags of tokens were not used as a feature but the others such as context word features, suffix, and prefix were used. The performance of the system as mentioned in Table 5.3. Its F-measure is reduced by 3.77% from F-measure of scenario one which is used as baseline.

Table 5.3. Performance of the ANER system without part of speech tags (scenario two)

Recall	71.87%
Precision	67.65%
F-measure	69.70%

In the third scenario, all the features except prefix were used and its F-measure is 74.61% as mentioned in Table 5.4. In this case, the F-measure is increased by 1.14% from F-measure of baseline. This indicates prefix has degrading effect. As we can see from table 5.2 and 5.4, performances of scenario one and three, the Recall value is the same but there is a difference in precision. This indicates for This might need further research why this feature has degrading effect.

Table5.4. Performance of the ANER system without prefix (third scenario)

Recall	75.00%
Precision	74.22%
F-measure	74.61%

Lastly, all the features except suffix were used and the performances of the system is F=70.65% as mentioned in Table 5.5. In this experiment the F-measure is reduced by 2.82% from the baseline. This indicates suffix has a positive contribution.

Table5.5. Performance of the ANER system without suffix (fourth scenario)

Recall	73.96%
Precision	67.62%
F-measure	70.65%

5.4 Discussion

The experimental results show that for different combinations of features, we have got different results. Table 5.6 shows the details of all the results.

Table 5.6. Details of all the experimental results.

Measurements[2]	All combination of features (scenario one)	Scenario two (without POS)	Scenario three (without prefix)	Scenario four (without suffix)
TP	72	69	72	71
FP	28	33	25	34
FN	24	27	24	25
Total	124	129	121	130
Recall	75.00%	71.87%	75.00%	73.96%
Precision	72.00%	67.65%	74.22%	67.62%
F-measure	73.47%	69.70%	74.61%	70.65%

From the above table, the maximum result is 74.61% which is the experiment without prefix (third scenario) and the minimum is 69.70% which is the experiment without POS tags of tokens (second scenario). To make detail discussion on the experimental results, first let's put the above table in chart form considering TP, FP, FN, and Total to see clearly where the difference comes.

[2] TP=True Positive, FP=False Positive, FN=False Negative

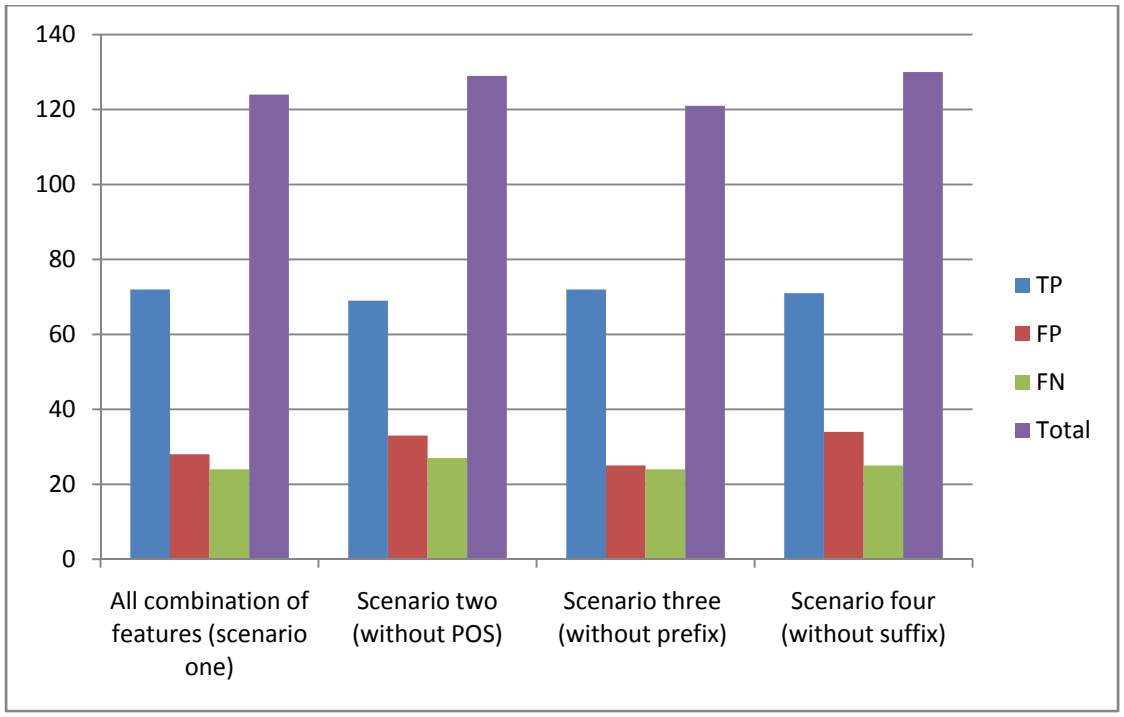

Figure 5.1: Experimental results for TP, FP, FN, and Total.

As we can see from the chart, the maximum value of TP is 72 which is scenario one and three. This indicates, there is no difference in TP value for both scenarios but there is a difference in FP values of the two scenarios. FP value of scenario one is greater than scenario three. This indicates detected tokens that are not in the reference (answer file) are decreased in scenario three as a result we have got better precision value and then we have got better F-measure value. This could be due to the fact that the prefix features extracted from the training corpus might not be enough and good features to recognize NEs. This might be due to small size of the corpus that we have used for training.

To compare Recall, Precision and F-measure values of all, let's see the chart in Figure 5.2. From the chart, Scenario one and three has the maximum Recall value but scenario two has the least. In a similar fashion scenario three has maximum Precision value but scenario four has the least. When we compare the F-measure of all the three, scenario three has the highest F-measure value. This is due to the reason mentioned above. But we can see that POS tags of tokens and suffix are important features of NER system.

In general, based on this experiment, all the features mentioned above except prefix gave the best combination of features for Amharic named entity recognition system.

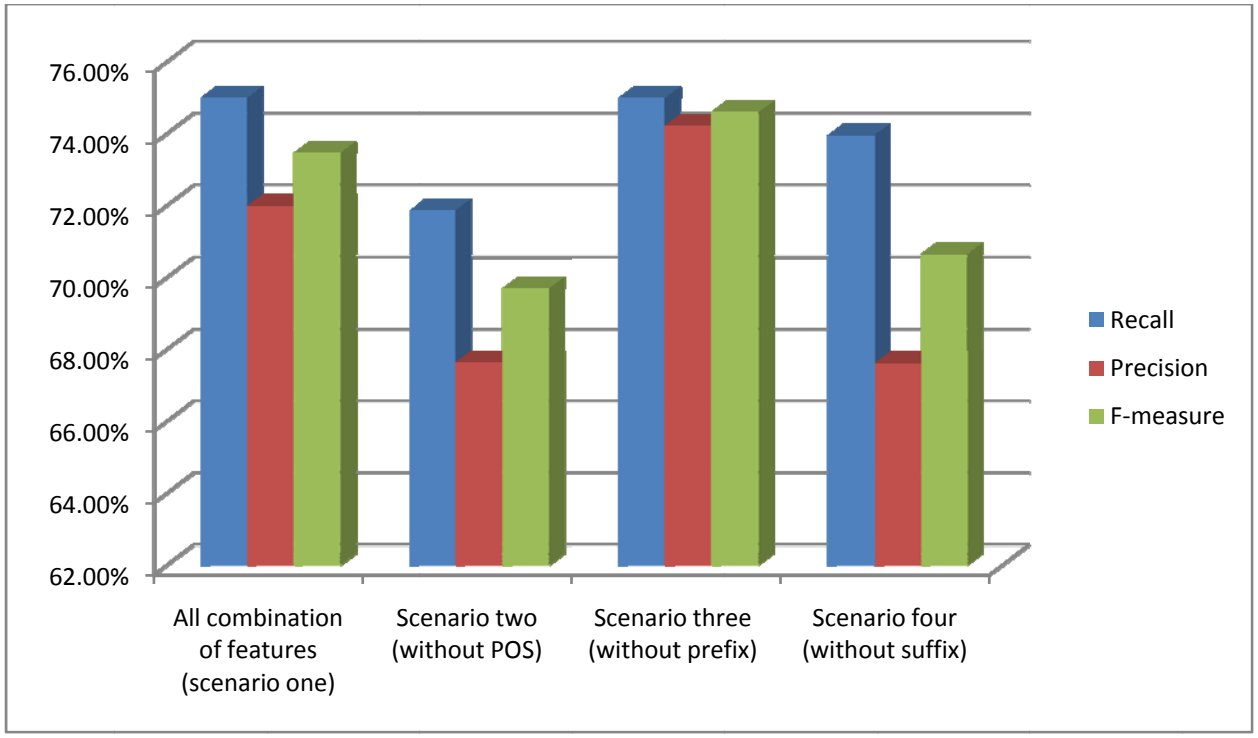

Figure 5.2: Experimental results for Recall, Precision, and F-measure.

Chapter Six

Conclusion and Future works

6.1 Conclusion

This research has presented CRFs based named entity recognition for Amharic language. The CRFs based method, as discussed in chapter two, is a statistical method which can be used for recognizing names for the given Amharic input texts.

The objective of the research was to test the applicability of the CRF based Named Entity Recognition System (NERS) for Amharic language. To do this research, 10,405 tokens were taken from WIC corpus and tagged manually with tags as mentioned in section 5.3. Some preprocessing tools like POS tagger for Amharic were trained and its model was used to tag part of speech of tokens since it is used as a major feature to recognize names. This POS tagger has a total accuracy of 81.555%.

During the experiments, four different scenarios were taken. In the first scenario, all the features mentioned in section 5.1 were considered and has got 73.47% and this was taken as a base line to compare for the other scenarios. In the second scenario, all the features except part of speech tags of tokens, were considered and has got a performance of 69.70%. Third and fourth scenario, all the features except prefix and suffix respectively, were considered and have got the performance of 74.61% and 70.65% respectively.

From the experiment, we made a conclusion that POS tags of tokens and suffix are important features to recognize NEs. In general, all the features mentioned above except prefix gave the best combination of features for Amharic named entity recognition system.

6.2 Future works

The study has shown that Amharic named entity recognition can be done automatically using conditional random fields algorithm. However, further research and developmental effort is needed to apply conditional random field approach in a full-fledged ANER system. Additional tasks that can be added to increase the performance of the proposed ANER system and future research directions are outlined below:

- With regard to corpus, Amharic language didn't have corpus which is manually tagged with named entity categories. For this study, small amount of corpus is tagged manually but this corpus is insufficient and very small. So, in the future huge amount of corpus for Amharic named entity recognition has to be prepared and the system has to be trained on that to improve its performance. In addition, we need to check the contribution of prefix feature to recognize NEs.
- Conducting this study with hybrid approach i.e. incorporating a rule based with CRF based approach is recommended since it might give a better performance.
- Conducting further study why prefix has degrading effect and why POS tagger is the best feature of all.
- Since the study covers some of the named entities such as location, organization and person, so other researchers can continue this study by incorporate the other named entities like numerical values such as numbers, percentages, and monetary values and date and times.
- This system is at thesis level then other researcher can develop a full-fledged Amharic Named Entity Recognition system as a project.

References

[1]. http://www.cs.bgu.ac.il/~nlpproj/hebrewNER/index.html, "Hebrew Named Entity Recognition (NER)", last accessed August 1, 2009.

[2]. Asif Ekbal, et.al. (2008). Bengali Named Entity Recognition Using Support Vector Machine. In Proceedings of IJCNLP-08 workshop on NER for South East Asian Languages, pages 51-58, Hydrabad, India

[3]. Mónica Marrero, et.al. (2009). Advances in Computational Linguistics. Research in Computing Science 41, 2009, pp. 47-58. ©A.Gelbukh (Ed.).

[4]. Kesarin Phanarangsan, et.al. (2006). Simple Named Entity Guidelines for Less Commonly Taught Languages. Linguistic Data Consortium – LCTL Team.

[5]. Mark Steven, et.al. (2000). Using Corpus-Derived Name List for Named Entity Recognition. Proceedings of the sixth conference on Applied natural language processing, Association for Computational Linguistics Morristown, NJ, USA.

[6]. Seid Muhie Yimam (2009). Amharic Question Answering System for Factoid Questions. Master's thesis, Addis Ababa University, Ethiopia.

[7]. http://www.umiacs.umd.edu/~nmadnani/pdf/crossroads.pdf, "Getting Started on Natural language Processing with Python" last accessed August 1,2009.

[8]. Yassine Benajiba, et.al. (2008). Arabic Named Entity Recognition using Optimized Feature Sets. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 284–293, Honolulu. Association for Computational Linguistics.

[9]. Rohini Srihari, et.al. (2000). A Hybrid Approach for Named Entity and SubType Tagging. Proceedings of the sixth conference on applied natural language processing. Seattle, Washington. Pages: 247 – 254.

[10]. GuoDong Zhou, et.al. (2002). Named Entity Recognition using an HMM-based Chunk Tagger. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 473-480.

[11]. Yuanyong Feng, et.al. (2006). Chinese Word Segmentation and Named Entity Recognition Based on Conditional Random Fields Models. Proceedings of the Fifth SIGHAN

Workshop on Chinese Language Processing, pages 181–184, Sydney, Association for Computational Linguistics.

[12]. Yassin Benajiba, et.al. (2008). Arabic Named Entity Recognition using Conditional Random Fields. Proceedings of 2008 Arabic Language and Local Languages Processing Workshop, LREC'08.

[13]. http://en.wikipedia.org/wiki/Information_retrieval, "Information retrieval" last accessed Oct.2/2009.

[14]. Liddy, E. D. In Encyclopedia of Library and Information Science, 2nd Ed. Marcel Decker, Inc (Natural Language Processing in Information Retrieval. Thorsten Brants. Google Inc..)

[15]. http://cs.nyu.edu/~sekine/papers/lrec02nova.pdf, "Summarization System Integrated with Named Entity Tagging and IE pattern Discovery" last accessed Oct.2/2009.

[16]. http://cidoc.mediahost.org/what_is_coref.pdf, "Information Extraction, Automatic" last accessed Oct.3/2009.

[17]. http://en.wikipedia.org/wiki/Dialog_system, "Dialog system" last accessed Oct.3/2009

[18]. http://corpora.amharic.org/resources/tagged-corpora/wic-tagged-news-corpus/, "Walta Information Center - Tagged Amharic News Corpus" last accessed July 10, 2010.

[19]. Fredrik Olsson, (2008). Bootstrapping Named Entity Annotation by Means of Active Machine Learning. Doctoral thesis, Swedish Institute of Computer Science AB. Gothenburg, Swedish.

[20]. Khaled Shaalan, et.al. (2007). Person Name Entity Recognition for Arabic. Proceedings of the 5th Workshop on Important Unresolved Matters, pages 17–24, Prague, Czech Republic, Association for Computational Linguistics.

[21]. Animesh Nayan, et.al. (2008). Named Entity recognition for Indian Languages. Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, pages 33–40, Hyderabad, India, Asian Federation of Natural Language Processing.

[22]. Asif Ekbal, et.al. (2008). Language Independent Named Entity Recognition in Indian Languages. Proceedings of the IJCNLP-08 Workshop on NER for South and South East

Asian Languages, pages 33–40, Hyderabad, India, Asian Federation of Natural Language Processing.

[23]. David Nadeau, et.al. (2007). A Survey of Named Entity Recognition and Classification. National Research Council Canada. Lingvisticae Investigationes. Volume 30, Edition 1. 23 pages.

[24]. Chowdhury, et.al. (2003). Natural language processing. Annual Review of Information Science and Technology, 37: 51–89. doi: 10.1002/aris.1440370103.

[25]. Simoes, et.al. (2009). Information extraction tasks: a survey (INESC{ID technical report No. 37/2009). Lisbon, Portugal.

[26]. P Srikanth, et.al. (2008).Named Entity Recognition for Telugu. Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages , pages 41–50. ©2008 Asian Federation of Natural Language Processing. Hyderabad, India.

[27]. Raymond J. Mooney, et.al. (2005). Mining Knowledge from Text Using Information Extraction. Volume 7, Issue 1, pages 3-10, ISSN:1931-0145. Association for Computing Machinery (ACM), New York, NY, USA.

[28]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.8785&rep=rep1&type=pdf, "Information Extraction, Automatic" last accessed August 25, 2010.

[29]. Pramod Kumar Gupta, et.al. (2009). An Approach for Named Entity Recognition System for Hindi: An Experimental Study. Proceedings of ASCNT-2009,CDAC, Noida, India, p. 103-108.

[30]. Yonatan Aumann , et.al. (2006). Visual Information Extraction. Knowledge and Information Systems archive. Volume 10 , Issue1, Pages:1 - 15 , ISSN:0219-1377. Springer-Verlag New York, Inc. New York, NY, US.

[31]. Roman Klinger, et.al. (2007). Classical Probabilistic Models and Conditional Random Fields. Algorithm Engineering Report, TR07-2-013, ISSN 1864-4503. Dortmund University of Technology, Department of Computer Science, Germany.

[32]. Xiaojin Zhu. (2007). CS838-1 Advanced NLP: Conditional Random Fields.

[33]. Atelach Alemu Argaw, et.al. (2007). An Amharic Stemmer : Reducing Words to their Citation Forms. Proceedings of the 5th Workshop on Important Unresolved Matters, pages 104–110, Prague, Czech Republic,. @2007 Association for Computational Linguistics.

[34]. Samuel Eyassu, et.al. (2005). Classifying Amharic News Text Using Self Organizing Maps. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 71–78, Ann Arbor. @2005 Association for Computational Linguistics.

[35]. Bethelhem Mengistu. (2002). N-gram-Based Automatic Indexing for Amharic Text. Msc thesis, Addis Ababa University, Ethiopia.

[36]. http://www.ethiopedia.com/index.php?title=Amharic, "Amharic" last accessed Oct.3/2009.

[37]. ጌታሁን አማረ. (1989 E.C). ዘመናዊ የአማርኛ በቀላል አቀራረብ . ንግድ ማተሚያ ድርጅት

[38].ተክለማሪያም ፋንታየ. "ኖሃተ ጥበብ ዘስነ ፀሁፍ". Central Printing Press, Addis Ababa.

[39]. Erik F., et.al. (2002). Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. International Conference On Computational Linguistics, Proceedings of the 6th conference on Natural language learning, Volume-20.

[40]. Alireza Mansouri, et.al. (2008). Named Entity Recognition Using a New Fuzzy Support Vector Machine. IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.2.

[41]. Dan Roth, et.al. (2005). Integer Linear Programming Inference for Conditional Random Fields. Proceedings of the 22nd International Conference on Machine Learning (ICML-2005), Bonn, Germany. Association for Computing Machinery, Inc.

[42]. Girma Awgichew Demeke, et.al. (2006). Manual Annotation of Amharic News Items with Part-of-Speech Tags and its Challenges. Ethiopian Languages Research Center of Addis Ababa University.

Appendix -1: The Amharic Character Set [35].

Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኲ	ኳ	ኴ	ኵ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ					
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ᳲ	ᳲ᳭	ᳲ᳡	ᳲ᳢	ᳲ᳣	ᳲ᳤	ᳲ᳥

Declaration

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

_____

MOGES AHMED MEHAMED

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

_____

SEBSIBE HAILEMARIAM (PhD)

Addis Ababa, Ethiopia

November 2010