ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

A Named Entity Recognition for Amharic

By

Besufikad Alemu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE:

Addis Ababa, Ethiopia

June, 2013

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

A Named Entity Recognition for Amharic

By

Besufikad Alemu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE:

Addis Ababa , Ethiopia

June, 2013

DECLARATION

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

_____

Besufikad Alemu

June, 2013

Thesis has been submitted for examination with our approval as University advisor.

_____

Solomon Teferra (PHD)

June, 20

Acknowledgement

First and foremost, I would like to thank God, who makes everything possible. I would extend my sincere gratitude to my advisor Dr. Solomon Teferra for his critical comments and for being my driving force throughout this work.

My special thanks go to my sister, Yilfashewa (Beti), for making all those hardships I had to go through easier. Beti, you are the greatest sister, mom and best friend. I could not have been able to make it without you.

I would also thank Moges Ahmed and my friends for their understanding and support.

Contents

List of table

List of figure

Abbreviation

MUC :	Message Understanding Conference
IE :	Information Extraction
NERC:	Named Entity Recognition and Classification
ORG:	Organizations name
PER :	Persons name
LOC :	Locations name
CONLL :	Conference on Computational Natural Language Learning
NER :	Named Entity Recognition
POS:	Part of Speech tag
NERs:	Named Entity Recognition System
NLP :	Natural Language Processing
ANER :	Amharic Named Entity Recognition
CRF :	Conditional Random Fields
IE :	Information Extraction
U.S.:	United States
SL :	Supervised Learning
HMM :	Hidden Markov Models
ME :	Maximum Entropy
SVM :	Support Vector Machines
UL :	unsupervised learning
NLTK :	Natural language tool kit
EM :	Expectation Maximization
SSL :	Semi- supervised Learning

Abstract

This thesis describes the development of Named Entity Recognition (NER) system for Amharic. 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. Amharic Named Entity Recognition (ANER) is proposed based on supervised machine learning, Conditional Random Fields (CRF). Amharic corpus of size 13,538 words have been developed with Stanford tagging scheme.

Since the objective of the work is working on feature set of Amharic Named Entities (NE), fifteen experiments were conducted by interchanging different features. Previous and next word, named entity tag of a word, word pairs, word shape, prefix and suffix features have been used to identify three NEs, Person, Organization and Location. The results of the experiment show that proper features combination in NER has a great role. The highest F-measure achieved in this work is 80.66%, with a window size of two on both(left and right)sides, previous and next tag of a current word and prefix and suffix with length four. The worst performance achieved is 61.97%, feature sets are a window size of two from the left side of a word and previous and next word of the current token. Finally the identified optimal feature sets from the experiment are: prefix and suffix with a length of four, previous and next NE tag of a token and window size features

Key word: Named Entity Recognition, Amharic Named Entity Recognition, Conditional Random Fields, Named Entities

Chapter One

Introduction

1.1. Background

According to (Grishman and Sundheim ,1996) as they are cited on [1], the term "Named Entity", now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC-6). At that time, MUC was focusing on Information Extraction (IE) tasks where structured information of company activities and defense related activities is extracted from unstructured text, such as newspaper articles. In defining the task, people noticed that it is essential to recognize information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions. Identifying references to these entities in text was recognized as one of the important sub-tasks of IE and was called "Named Entity Recognition and Classification (NERC)" [1].

Named entities are phrases that contain the names of persons, organizations and locations. Example:

[ORG AAU] Official [PER Besufikad] heads for [LOC Addis Ababa] .

This sentence contains three named entities: Besufikad is a person, AAU is an organization and Addis Ababa is a location. Named entity recognition is an important task of information extraction systems. There has been a lot of work on named entity recognition, especially for English [2].

According to [1], a good proportion of work in NER research is devoted to the study of English but a possibly larger proportion addresses language independence and multilingualism problems. German is well studied in CONLL-2003 and in earlier works. Similarly, Spanish and Dutch are strongly represented, boosted by a major devoted conference: CONLL-2002. Japanese has been studied in the MUC-6 conference, in addition to the above mentioned

languages he has listed many European languages that have been tried to develop a good NER.

1.2. The challenge of Named Entity Recognition

The ambiguity of NEs in text resides at two different levels; detection and classification. The first involves disambiguating NEs from non NEs in text and the second involves classifying them into classes e.g. person or location. The two processes of NER are sequential and the correctness of the identification phase is a prerequisite for the classification phase. The second level is dependent on the first one; NEs will not be classified correctly if they are not detected correctly in the first place [3].

The level of ambiguity that resides at these two levels differs from one language or domain to another. In English text, for instance, the identification would normally rely on orthographic case information (lower and upper) to detect NEs. If we manage to identify all NEs using that information then the fundamental problem is classifying them. For the classification, assuming that we managed to collect and group all NEs into lists for each NE class, there still would be cases where one token might fall into more than one list [3]. For instance, in Amharic the word *Amede* could be a person name or location. According to [3], Thus more information is required to classify NEs correctly; for example, the context in which the word occurs, such as the previous word. Thus, if the word *Amede* is preceded by *Ato* then it is more likely to be a person name. Titles and designators have proved successful in NER and are called triggering words or triggers.

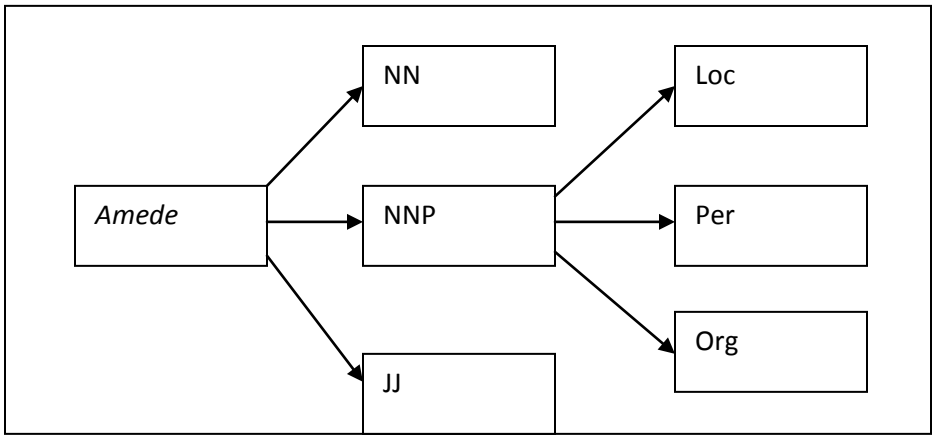

Figure 2.1 State- of- a sentence ambiguity example Shabib (2011) [1]

To be able to identify NEs successfully in such a situation, there would be a need to analyze the context, given that each word category would have a different context. For instance, an adjective typically cannot follow a verb without a determiner and also noun typically cannot be followed by an adjective in English. Thus, analyzing the surrounding lexical and POS context is crucial. If we have the means to generate the POS information correctly, then the problem would be solved to a large extent. However, it is not feasible as POS taggers rely on case information to tag proper nouns [3].

[1] proper noun (NNP), noun(NN), adjective (JJ), Location (L), person name(per), and organization (Org)

1.3 Statement of the problem

The motivation that initiated the researcher to conduct this research on NER is the advantageous and application areas of NERs. As it is stated in [4], NER is a key technology of Information Extraction and Open-Domain Question Answering.

According to [5] , Amharic 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 is used nationwide as the official language of the nation. Despite having large speaker population, the language has little computational linguistic resources. So, conducting a research on such kind of language is advisable to overcome the current shortage of computational linguistic resources.

Amharic is one of the under-resourced languages. Hence, there is no full-fledged Amharic Named Entity Recognition system that could contribute towards designing NLP. So that conducting Amharic NER with best performance will have a contribution to advance research in NLP for Amharic language.

There is only one attempt on Amharic named entity recognition. And further improvement regarding Amharic NER(ANER) has never been tried after the first ANER [6] attempt. Nationwide also there have been limited attempts for developing a NER of Ethiopian languages, Amharic by Moges[6] , Afan Oromo by Habtamu [7] and Mandefro [8] were attempted to develop Named Entity Recognitions. Moges [6] shown that some features played a great role in changing the performance of the Amharic NER system. So, in the proposed system the researcher tried to reconsider features that could increase the accuracy of Amharic NER.

According to [9] the best choice of feature combination chosen via the validation set correlated perfectly with the choice that has the best test set

performance, and in a system like [6] which followed Conditional Random Field (CRF) model, well-chosen features along with dictionary-based features tend to improve the CRF model's performance [10]**.** These shows considering other features would help to achieve a better performance. Therefore it is the present research to explore best feature sets for designing a high performance Amharic Named Entity Recognition.

Hence the research questions to be addressed have been the following

- What are the best feature sets for enhancing the performance of Amharic NER?
- Does a combination of different feature have an effect on the system?
- Which features are the optimal features sets in ANER?

1.4 Objectives of the study

General and specific objectives that have been tried to achieved are:

1.4.1 General objective

The general objective of this research is identifying the optimal feature sets that improve the performance of ANER system.

1.4.2 Specific objective

- Review literature so as to understand the role of named entity and approaches in NER
- Identify as many features as we can.
- Identify the optimal combination of the features that brings the best ANER Performance
- Selecting features that could be integrated with Amharic NER.

- Compare models that could help to improve the performance of Amharic NER.
- Evaluate performance of the proposed system by using a test dataset

1 .5 Significance of the study

NER was successfully incorporated into name searching systems to identify names in queries and the underlying data source for information retrieval [11]. Another important application is question answering, which was driven by one of Text Retrieval Conference's tracks. According to (Noguera et al. 2005) as cited by [3] NER is a core component of these systems and the effect of NER was measured in. The idea was to consider only documents that have at least one named entity provided in the query. It was found that NER could reduce the data returned by the IR system by 62% without any loss of information and by 92% with acceptable loss. Detecting NEs in a query was also success [3].

Conducting NER for Amharic is also very important for the development of Natural Language Processing (NLP)(such as, Speech synthesis, speech recognition, Question Answering, Information retrieval, Information extraction) [12]. In order to achieve such kind of advantages, developing a system with a high performance is unquestionable and to have a high performance ANER, finding the optimal features and identifying the best features combination are the main task of NERs.

Result of this research could be an input for different disciplines of NLP specifically in Amharic. This includes, Information extraction, Question answering, Ontology construction etc.

1.6 Scope of the study

Most common among marked categories are names of people, organizations and locations as well as temporal and numeric expression [13]. And within each category there could be additional sub categories. For example, for location may have city, mountain, river, sea, village etc. as its subcategory. In this study due

to time constraint only the three main NE classes names of people, organization and locations without their subcategories were taken.

1.7 Methodology

NER is a compound task that is to be done by different components at different steps. The research also has different developmental stages. First finding out the corpus from a news source was researchers' initial task. The learning methodology to be adopted requires customization, so it has its own computational task. Selecting the appropriate tools and techniques were the parallel task during all the stages.

1.7.1 Literature review

This work is conducted by reviewing a number of related literatures from books, journal articles, conference proceeding and the internet, so that choosing appropriate tools and methods were supervised. The large portions of reviewed materials are conference and journal articles.

1.7.2 Tools

A tokenizer during preprocessing of both training and testing data was developed by using Python programming languages. The reason python was chosen by the researchers was, it is easy for text processing. For NER part, Stanford NER tool is used.

As [12] describes machine learning techniques which can handle multiple features, a machine learning approach was analysed as an approach for NE learning process.

1.7.3 Data source and preparation

As it is mentioned above the tool was tested with freely available Amharic texts, which are collected from WALTA. Due to the availability of transliterated articles and acquiring those articles for freely, WALTA is chosen.

The collected articles were further edited manually by removing those sentences which does not have NEs at all. The resulting data is then tokenized and tagged manually in accordance with the Stanford standard. Finally, NE corpus of size 13538 was developed.

1.7.4 Test Procedure

The performance of methods and tools are measured based on the three main scoring approaches: precision, recall and F-measure. According to [13], these measures are the most commonly used for the assessment of statistical extraction systems and trace their origins back to the information retrieval discipline.

1.8 Organization of the work

The rest of this thesis is organized as follows. Chapter 2, literature review, concept related to named entity recognition is discussed. In the chapter background information of information extraction, feature sets types, machine learning approaches, CRF model and nature of Amharic language are described. In chapter 3 local and global literatures are reviewed and their achievements are discussed. chapter 4 contains design and implementation part of this work, experimentations that are conducted in the research are found in this chapter. Finally, conclusion and recommendations which are given based on the experimentation are found in chapter 5.

Chapter two

Literature review

2.1 Information Extraction (IE)

Currently, there is a considerable interest in using these technologies for information retrieval, since there is an increasing need to localize precise information in documents, for instance, as the answer to a question, rather than retrieving the entire document or a list of documents.

According to [14],IE systems extract domain-specific information from natural language text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies, and physical objects. Domain-specific extraction patterns (or something similar) are used to identify relevant information.

This early definition on information extraction is too limited and this definition is the same as current named entity extraction definition, Information extraction system should not necessarily be domain specific rather it should be domain independent or at least flexible to any domain with a minimum amount of engineering effort.

Some scholars are saying information extraction only extract a fragment information but other says it is the process of extracting fragment information from a corpus and then process it in a way that it gives meaning. According to [15],IE isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework. The goal of information extraction research is to build systems that find and link relevant information while ignoring extraneous and irrelevant information.

As [16] defines IE is the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making the information more suitable for information processing tasks.

2.2 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. For the first time the name NER system coined at MUC-6 but the system was applied in different names those are domain specific (extract and recognize company names). The term NER came after the increasing need of extracting specific information from a large corpus. To conduct a NER system defining the right tag for an entity is the most important task, there are conventions those defined tagging techniques of a named entity in a corpus. Message Understanding Conferences-6(MUC-6) and MUC-7 and Conference on Computational Natural Language Learning (CoNLL) are among the conventions that most researchers who are conducting on NER are currently using [17].

2.2.1 Message Understanding Conferences (MUC)

The Message Understanding Conferences (MUC) was a series of conferences organized by the (U.S.) National Institute of Standards and Technology and the U.S. Department of Defense Advanced Research Projects Agency. These events were held between 1987 and 1998 as part of the TIPSTER Text program [3] . It is believed that MUC seven conferences on IE contributed a lot for the development of NLP research particularly in IE. The MUC conferences focused on English, Chinese, Japanese and Spanish languages the three Identified MUC conference entities are ENMAX, TIMEX and NUMEX.

1. **ENMAX** contains, **Person name** - named person or family. Eg. Besufikad ,Alemu etc.

   Location name - name of politically or geographically defined location. Eg, Addis Ababa, Ethiopia etc.

   Organization name - named corporate governmental or other organizational entity. Eg. Addis Ababa University, International Leadership Institution etc.

2. **TIMEX** contains, **DATE** includes complete or partial date expression. ሃሙስ ጠዋት/Thursday Morning/ መጋቢት ፲፫ / March 22/ etc.

TIME includes complete or partial expression of time of day ቅዳሜ ማታ / Saturday night/

3. **NUMEX** contains- numeric expressions, monetary expressions and percentages; expressed in either numeric or alphabetic form.

   Money- monetary expression eg. 5 birr, $3 etc.

   Percent – percentage expression eg. 5%, 100% etc.

From the above listed MUC conference entity types researchers conducted by using either partially or complete availability of the types.

2.2.2 Conference on Computational Natural Language Learning

CoNLL is highly focused on the named entity recognition task. The named entity types classification somehow differs from MUC, the three sub-classes in MUC first class (NAMEX) are treated as an independent class and by introducing Miscellaneous names which includes product names, names of services, etc [18]. CoNLL categorized named entity types in four groups. In contrast to MUC the languages considered by CoNLL were Spanish and Dutch in 2002 and German and English in 2003.

2.3 Machine Learning

As (Grobelnik M and Mladeni´c D ,2005) cited on [19]different knowledge discovery methods have been adopted for the problem of semi-automated ontology construction including unsupervised, semi-supervised and supervised learning over a collection of text documents, using natural language processing to obtain semantic graph of a document, visualization of documents, information extraction to find relevant concepts, visualization of context of named entities in a document collection.

Machine learning algorithms can be classified as Supervised, Unsupervised and semi-supervised. The distinction comes from how the learner classifies data. In this section we will see the three kinds of machine learning.

2.3.1 Supervised learning (SL)

Supervised learning, the classes are predetermined. The classes are previously known. In a supervised learning, let the domain of instances be X and the domain of labels be Y. Let P(x, y) be an unknown joint probability distribution on instances and labels X × Y. Given a training sample $\{(x_i, y_i)\}$, supervised learning trains a function f : X → Y in some function family F, with the goal that f (x) predicts the true label y on future data x [20].

According to [21]the goal of supervised learning is to learn a mapping from x to y, given a training set made of pairs (x_i, y_i). Here, the $y_i \in Y$ are called the labels or targets of the examples x_i.

The process in supervised learning is that a certain part of data will be labeled with known classifications. The machine learner's task is to search for patterns and construct mathematical models. These models then are evaluated on the basis of their predictive capacity in relation to measures of variance in the data itself.

Supervised learning is the current dominant technique for addressing the NER. SL techniques include Hidden Markov Models (HMM), Decision Trees, Maximum

Entropy Models (ME), Support Vector Machines (SVM) and Conditional Random Fields (CRF). These are all variants of the SL approach, which typically feature a system that reads a large annotated corpus, memorizes lists of entities, and creates disambiguation rules based on discriminative features [22].

2.3.2 Unsupervised Learning (UL)

Unsupervised learning algorithms work on a training sample with n instances $\{\mathbf{x}_i\}_{i=1}^{n}$. There is no teacher providing supervision as to how individual instances should be handled, Common unsupervised learning tasks include, clustering, where the goal is to separate the n instances into groups, novelty detection, which identifies the few instances that are very different from the majority and dimensionality reduction, which aims to represent each instance with a lower dimensional feature vector while maintaining key characteristics of the training sample [20].

In unsupervised learning, there is no such supervisor and we only have input data. The aim is to find the regularities in the input. There is a structure to the input space such that certain patterns occur more often than others, and we want to see what generally happens and what does not [23].

According to [23] the goal of unsupervised learning is to group data into clusters . In fact, the basic task of unsupervised learning is to develop classification labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group. These groups are termed clusters, and there is a whole family of clustering machine learning techniques.

One can try to gather NEs from clustered groups based on context similarity. There are also other unsupervised methods. Basically, the techniques rely on lexical resources (e.g., WordNet), on lexical patterns, and on statistics computed on a large unannotated corpus [22].

In unsupervised, the machine is not told how the texts are grouped. Its task is to arrive at some grouping of the data. Unsupervised learners are not provided with classifications.

2.3.3 Semi- supervised Learning (SSL)

Semi-supervised machine learning came up with the combination of supervised and unsupervised learning. As [21]defines, SSL is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information but not necessarily for all examples. Often, this information will be the targets associated with some of the examples. In such kind of cases , the data set $X = (x_i)_{i \in [n]}$ can be divided into two parts: the points $X_l := (x_1, \ldots, x_l)$, for which labels $Y_l := (y_1, \ldots, y_l)$ are provided, and the points $X_u := (x_{l+1}, \ldots, x_{l+u})$, the labels of which are not known, and he called this a standard semi- supervised learning.

According to [20],SSl is not a half way rather it is an extension of SL and SSl also known as classification with labeled and unlabeled data (or partially labeled data), this is an extension to the supervised classification problem. The training data consists of both l labeled instances $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabeled instances $\{x_j\}_{j=l+1}^{l+u}$. One typically assumes that there is much more unlabeled data than labeled data, i.e., $u > l$. The goal of semi-supervised classification is to train a classifier f from both the labeled and unlabeled data, such that it is better than the supervised classifier trained on the labeled data alone [20].

2.4 Feature set 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. Authors supported the above idea with an 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 [7].

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”>”

But the above definition is working only for European language those are following capitalization to named entities (i.e. English), where as for other languages like Amharic, Hindu, Arabic …etc the above mentioned feature vector did not work.

One of the most challenging aspects in machine learning approaches to NLP problems is deciding on the optimal feature sets, there are different feature set and selecting the best features is not easy. In the baseline research (Moges 2010) the features have shown that a change and selection of them also was not easy. So, as [24] described when developing a NER for a specific language, having a large number of features at the beginning is advisable and it helps to come up with the optimal feature sets.

Different authors classified feature set in different ways, some grouped local knowledge and global knowledge [25],within each group there is also subgroups ;and others [24, 26] categorized the sub groups independently .

According to [25] local knowledge are feature which can be extracted from a token (word) being labeled and its surrounding context.

Token (word) – based features contains suffixes, prefixes and orthographic features (shape) of the current token. Context – based features are Previous and next words of a particular word. The performance we get from token based and context based features are different. They are also domain specific, as [25]describes context based feature outperform than word (token) in biomedical entities and in newswire the result is vice versa.

External knowledge features are additional features that determine the entity of a token beyond the context and word based feature. According to [25] External knowledge features are part-of-speech (POS) tags, shallow parsing and gazetteers.

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 [7]. Pos tagging by itself is challenging and time consuming due to this the baseline and other NERs are used manually tagged corpus. Because of the aforementioned reasons [27]developed without it.

Phrasal clustering- is clustering a corpus in to n clusters by using different clustering techniques and use the phrase to identify tokens entity.

Gazetteers- list of predefined names of persons, locations, organizations etc. Even though developing gazetteers is not easy working on the available list is also difficult. Different problems are raised from the misuse of gazetteers. According to [28] semantic ambiguities are occurred when we use gazetteers, there is a situation that a single name could be person name , organization name or location name. For example, *Alem* can be name of a person, organization or location. In such kind of scenario it is difficult to decide the right entity for a token.

The other challenge of using a well prepared gazetteer is that they are memory intensive that comes from people names and organization names are different due to this there will be numerous names in the list but the occurrence of those tokens in a corpus is rare sometimes they might not be presented once in the whole document and having those unused words in the corpus is wastage of our memory space. In addition to this a large coverage gazetteer would not guarantee a better accuracy. As [3]said this is due to the fact that person names are mostly found in form of nouns and adjectives. Whereas rearranging the list in the gazetteers could be improve the performance, according to [13] rearranging means selecting the most important out of the predefined list and adding the remained names that believed to bring a better performance.

2.5 Conditional Random Fields (CRF)

Conditional Random Fields can be understood as the sequence version of Maximum Entropy Models, that means they are also discriminative models [29]. The Conditional Random Fields (CRF) are probabilistic models for computing the probability p (Y|X) of a possible output Y = (y_1,…y_n) ε Y^n given the input X = (x_1 …x_n) ε X^n which is also called the observation. In this section CRF which is used in our ANER model is discussed in detail.

According to [30] linear chain CRFs is a special form of a CRF, which is structured as a linear chain that models the output variables as a sequence.

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

Where, j specifies the position in the input sequence x, but the weights λ_i are not dependent on the position j. This technique, known as parameter tying, is applied to ensure a specified set of variables to have the same value. n specifies the length of the sentence, m specifies the number of feature templates.

The normalization to [0; 1] is given by

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y}\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)\right) . \quad 2.$$

As it is indicated in the above two equations, features are dependent on the label sequence and herewith on the state transitions in the finite state automaton. So it is important to point out that only a subset of all features f_i is used in every transition in the model.

According to [30],the strategy to build a linear-chain CRF summarized in the following way:

1. Construct Stochastic finite state automaton (SFSA) S= (S, T) out of the set of states S (with transition T=$(s, \dot{s}) \in S^2$). It can be fully connected but it is also possible to forbid some transitions because the decision made in

this stage is depending on the training data and the transitions contained there.

2. Specify a set of features templates $F = \{g_1(\vec{x}, j), \ldots, g_h(\vec{x}, j)\}$ on the input sequence. This helps for the generating features f_i.

   $$g(x,j) = \begin{cases} 1 & if\ x_j = names \\ 0 & else \end{cases}$$

3. Generate set of features $\mathcal{F} = \{\forall s, \dot{s} \in S.\ \forall g_o \in F : f_k(s, \dot{s}, g_o)\}$

There are two problems in a linear chain CRFs that plays a great role in varying the performance of NERs.

The first one is, there is a given observation x and a CRF M: here the problem is finding the most probable label sequence Y. This problem is the most common application of a conditional random field to find a label sequence for an observation.

The other one is, given label sequences Y and observation sequences X: How to find parameters of a CRF M to maximize P (Y|X, M) are also a problem. This problem is a question of how to train to adjust the parameters of M which are especially the feature weights. In our system we have tried to address both of the above mentioned problems of linear chain CRF model.

2.5.1 CRF training

To train a CRF we need labeled data sequences. According to [31],CRFs training on unlabeled data is difficult. The labeled sequences are, { $(x^{(1)}, y^{(1)}),\ldots,(x^{(n)}, y^{(n)})$}, where $x^{(1)}=x_{1:N1}$ the first observation sequence, and so on. The main objective for parameter learning is to maximize the conditional likelihood of the training data.

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

$$= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)\right)}{\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right)} \right) \right] \quad 3.$$

During training there are features that are more than enough to determine the tag of a specific term. That means over fitting could occur and to control such kind of scenario over fitting likelihood is penalized with the term $-\sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma^2}$. The parameter δ^2 models the tradeoff between fitting exactly the observed feature frequencies and the squared norm of the weight vector. The smaller the values are, the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced. The notation of the likelihood function L(T) is reorganized as:

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

$$= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right) \right) - \right.$$

$$\left. - \log \left(\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right) \right) \right] - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

$$= \underbrace{\sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)}_{\mathcal{A}} -$$

$$\underbrace{- \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \log \underbrace{\left(\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right) \right)}_{Z_{\vec{\lambda}}(\vec{x})} - \underbrace{\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}}_{\mathcal{C}}}_{\mathcal{B}} . \quad 4.$$

The partial derivations of L(T) by the weights for parts A,B and C are computed separately.

Derivation for part A:

$$\frac{\partial}{\partial \lambda_k} \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{j=1}^{n} f_k\Big(y_{j-1}, y_j, \vec{x}, j\Big) \tag{5.}$$

Derivation for part B, which corresponds to the normalization, is given by:

$$\begin{aligned}
\frac{\partial}{\partial \lambda_k} \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \log Z_{\vec{\lambda}}(\vec{x}) &= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \frac{\partial Z_{\vec{\lambda}}(\vec{x})}{\partial \lambda_k} \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \sum_{\vec{y}\,'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\Big(y'_{j-1}, y'_j, \vec{x}, j\Big)\right) \cdot \\
&\qquad \cdot \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j) \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}\,'\in\mathcal{Y}} \underbrace{\frac{1}{Z_{\vec{\lambda}}(\vec{x})} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\Big(y'_{j-1}, y'_j, \vec{x}, j\Big)\right)}_{=p_{\vec{\lambda}}(\vec{y}|\vec{x})} \cdot \\
&\qquad \cdot \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j) \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}\in\mathcal{Y}} p_{\vec{\lambda}}(\vec{y}\,'|\vec{x}) \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j)
\end{aligned} \tag{6.}$$

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

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \tag{7.}$$

Equation 5, the derivation of part A, is the expected value under the empirical distribution of a feature f_i:

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \tag{8.}$$

Accordingly, equation 6, the derivation of part B, is the expectation under the model distribution:

$$E(f_i) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}'\in\mathcal{Y}} p_{\vec{\lambda}}(\vec{y}'|\vec{x}) \sum_{j=1}^{n} f_i(y'_{j-1}, y'_j, \vec{x}, j) \quad 9.$$

The partial derivations of L(T) can also be interpreted as

$$\frac{\partial \mathcal{L}(\mathcal{T})}{\partial \lambda_k} = \tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} \quad 10.$$

And the maximum likelihood method finds the parameter by equating the first derivation to 0 as:

$$\tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} = 0 \quad 11.$$

Computing $\tilde{E}(f_i)$ is easily done by counting how often each feature occurs in the training data. Computing $E(f_i)$ directly is impractical because of the high number of possible label sequences. 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 complex to explain in this study [31].

2.6 Named entity recognition tools

There are tools which helps to label NEs, Natural language tool kit (NLTK), Stanford named entity recognition, GATE, Lingpipe… etc. NLTK is a python platform that is more suitable for Part of Speech (POS) tagging. Stanford NER is a Java implementation of a Named Entity Recognizer developed by Stanford University. When Stanford started NER their scope was focused on identifying Biological terms (genes and proteins) from biomedical papers [32]. After their first attempt, they have expanded their scope by increasing the classes and also with regard to features the model they have used changed from time to time till

they implemented a sequence classifier to the unseen domains (i.e. CRF). We have used this customizable and open source NER tool for labeling our data.

Lingpipe is also 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 [33]. [6, 8] are used this tool.

2.7 Amharic language structure

In this section we have discussed language structure of Amharic with regard to our focused area, which is NE in Amharic. The section describes the background of Amharic script, word classes and characteristics of Amharic NEs. Even though, there are different word classes in Amharic we have only focused on Nouns and pronouns which are related to our work.

According to [5], 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

The present Amharic writing system was adopted from the Ge'ez writing system. Ge'ez, which belongs to the class of Semitic languages, was the language of literature in Ethiopia in earlier times [34]. The ancient Sabaean script is in turn attributed to the source of the Ge'ez script. However, as [34] explains, the number of symbols in the original Sabaean script and their shapes have changed into Ge'ez and then later on into Amharic. Moreover, some new

symbols have been added to Amharic. The current Amharic writing are combination of all Ge`ez fidel and some added characters.

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. It consists of 33 consonants, giving 7*33=231 syllable patterns, or fidels. In addition to the 231 characters, there are other non-standard alphabets which contain special features usually representing labialization. Each alphabet represents a consonant together with its vowel. The vowels are used to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order. Refer to appendix-1 for a complete list of the symbols [35].

Word classes of Amharic language are classified in to different categories by different scholars. As Mersehazen(1935 e.c), cited on [36] put the word classes as eight. Those are preposition, noun, conjunction, interjection, verb, adjective, pronoun and verb. Whereas, the recent works (Baye, 1987 and 97) as cited by [36], reduced the previous classes in to five word classes. The reduction of Amharic word classes are based on different reasons, Baye's reduction of Mersehazen's classification seems based on the role of words in syntax, which means by considering the clear role of words in Amharic grammar.

2.7.1 The Noun Class

Like English, Amharic nouns are words used to name or identify any of a class of things, people, places, organization or ideas or a particular one of these [37]. Formations of nouns are either simple, derivative or compounds.

As [34] Discusses, the formation of nouns are described as follows:

Simple forms; consisting of two, three, or four letters.

A. Bilateral.

a. Ending in the second order:

ሙሉ፡ full.

ጥሩ፡ good

ክፉ፡ bad

These all words are originating from or representing the passive participle.

b. Ending in the third order, generally signifying an agent:

መሪ ፡ guide

ሰፊ : wide

ሰሪ : workman

c. Ending in the fourth order:

ውኃ፡ water

ስጋ: flesh

ካራ: knife

d. Ending in the fifth order:

ቅቤ፡ butter, oil.

ጥሬ: genuine, original.

በሬ : ox.

e. Ending in the sixth order.

ቀን ፡ day

ላም : cow

ሙዝ፡ banana.

Most of bilateral nouns are ending in these order and they are numerous.

f. Ending in the seventh order:

ጉዞ ፡ a day's march.

ጎጆ ፡ small thatched house.

ቆሎ ፡ fried grain.

In triliteral and bilateral, [34] listed out different nouns. Noun words with more than two alphabets (fidel*)* are discussed in detail.[2]

Compound nouns: are nouns which are written as two separate words and sometimes as a single word [38]. E.g. " ቤተ-መንግስት" which mean palace. "ትምህርት ቤት" school "ፅህፈት ቤት" and "ፅፈትቤት" which both mean secretariat office.

Derivative nouns- are nouns which repeat themselves in order to pluralize the quantity they are expressing for. E.g.

ፍሬ	ፍሬ-ኣ- ፍሬ	ፍራፍሬ
ቁስ	ቁስ-ኣ-ቁስ	ቁሳቁስ
ትል	ትል-ኣ-ትል	ትላትል

2.7.2 Pronouns

According to [34] pronouns are divided as in other languages, into:

1. Personal
2. Possessive
3. Demonstrative
4. Relative
5. Interrogative
6. Reflective and

7. Distributive Pronouns.

[2] C. W. Isenberg, *Grammar of The Amharic Language*. London: Richard Watts 1976.

These classes are grouped into two categories, namely, Separable and Inseparable Pronouns.

Personal
Possessive
Demonstrative
Interrogative
} Separable

Relative
Distributive
} Inseparable

2.7.2.1 Separable Personal Pronouns

They are three for the Singular, and three for the plural. The singular has some peculiarities. The first Person has not the gender expressed, whereas, the second and third have distinct forms for the Masculine and for the Feminine gender. The second person has, besides, three distinctions of honor.

	Singular			Plural
	Masculine	common	Feminine	common
1 Person		እኔ: I		እኛ : we.
2 Person	አንተ: you		አንች: / አንቺ: } thou	እናንተ: you.
		አንቱ / እርስዎ } you		
3 Person.	እርሱ: he,it.		እርስዋ: she, it.	እርሳቸው: they.

The above structure is Amharic separable personal pronoun formation and Detailed discussion on Amharic pronouns are found in [34].

2.7.3 Amharic Named Entities

According to [39], NEs are phrases that contain the names of persons, organizations and locations. In the expression named entity, the word named restricts the task to those entities for which one or many rigid designators stands for the referent.

Person names (PERSON) - name of a person including his/her father and grandfather

Location name (LOCATION) – name of geographical entities like cities, regions, country.

Organization name (ORGANIZATION) – name of entities like companies, institutions and organizations.

Amharic has no case information like English and other Latin character languages which ease identification of NEs. It is one of those properties used in the tool (Stanford Named Entity Recognition), but due to the previously mentioned property of Amharic we have excluded capitalization feature.

There are words that mostly co-occurred with NEs, Clue words are useful properties in identifying NE and that mostly precede NE words and sometimes they succeeded NEs. Clue words for each NE classes are different, some of them for PERSON are:

ፕሬዚዳንት	President
አቶ	Mr.
ወ/ሮ	Mrs.

ፕሬዚዳንት ግርማ ወ/ጊዮርጊስ	President Girma W/Giorgis
አቶ በዓሉ ግርማ	Mr. Bealu Girma
ወ/ሮ አዜብ	Mrs. Azeb

Organization NEs are most of the time succeeded with the following clue words.

ካፒታል ሆቴል	Capital Hotel
ሐረማያ ዩኒቨርሲቲ	Haramaya University
ፕሬዚዳንት ቢሮ	Office of president

የተባበሩት መንግስታት ድርጅት United Nations Organization

Location NE classes have the same properties with Organization. In addition to clue words, Amharic NEs present with affixes.

Affixes are classified in to three, Prefix, Suffix and Infix.

Prefixes, suffixes and Infixes are sets of letters that are added to the beginning or end of another word. They are not words in their own right and cannot stand on their own in a sentence.

E.g. □□□□ □□□□□ □□□□□ □□□□□ □□□□

Esayas Afewerki is president of Eriteria.

□□□□□ □□□□□ □□□□□....

Esayas Afewrki's government…

On the first sentence /□□□□ □□□□□ / is a PERSON name, whereas, when we add the prefix /□/ the name changes it class to ORGANIZATION.

E.g. አዲስ አበባ ዩኒቨርሲቲ በኢትዮጵያ ዋና ከተማ ውስጥ ይገኛል፡፡

Addis Ababa University is found at the capital city of Ethiopia.

ከአዲስ አበባ ዩኒቨርሲቲ እስከ…

From Addis Ababa University….

Here in the first sentence the word /*አዲስ አበባ ዩኒቨርሲቲ* / is an ORGANIZATION name, but in the second form of the word /*ከአዲስ አበባ ዩኒቨርሲቲ* / becomes LOCATION name. The change comes after the addition of prefix /ከ/. N.B the second case is not always true, because the prefix by itself is not sufficient to identify NE classes. For eg.

አዲስ አበባ ዩኒቨርሲቲ በኢትዮጵያ ዋና ከተማ ውስጥ ይገኛል፡፡

Addis Ababa University is found at the capital city of Ethiopia.

ከአዲስ አበባ ዩኒቨርሲቲ የተማሪዎች መማክርት…

From the student council of Addis Ababa University…

In such cases the name/አዲስ አበባ ዩኒቨርሲቲ/ with prefix /ከ/ and the name /አዲስ አበባ ዩኒቨርሲቲ /in the first sentence have the same NE class, ORGANIZATION.

2.7.4 NER challenges for Amharic

Several studies indicated that NER is a difficult task and a number of challenges are still not addressed because of many reasons: Ambiguity, Agglutinative Nature of the Language, Spelling Variations and A Nested Named Entities**.**

2.7.4.1 Ambiguity

In Amharic there are proper names that have more than one NE class. ፀሐይ (*Sun)* can be person's name and it is used to name Sun. Amede, H/Giorgis, Haile Gebresilase ,Abune Petros and Belay Zeleke are some of the names that are used for both location and person's name in Amharic. During model creation, handling ambiguity words are performed with a feature that can learn their contextual pattern in the sentence.

2.7.4.2 Agglutinative Nature of the Language

Agglutinative property means that some additional features can be attached to the word to add more complex meaning. According to [7],most of the time NE root words cannot be found in a document, rather they are found in their morphological derivations that are formed by affixes. As we have discussed in section 2.7.3 Amharic Named Entities prefixes helps to identify class of a word in Amharic. For example, when the prefix, ከ- adds to ኢትዮጵያ (Ethiopia) it becomes ከኢትዮጵያ (From Ethiopia) and we can used such kind of affix as a feature to classify that word under LOCATION class.

2.7.4.3 Spelling Variations

In Amharic there are words which can be written in different formats. For instance, "ፀሐይ" and "ጸሐይ", "ፀሀይ" and "ጸሀይ" all mean the same, "sun" although they are written differently (indicating alternate use of ፀ and ጸ, and ሀ and ሐ. In our research we have tried to handle such kind of language challenges by using transliterated data, in transliterated corpus all the afore mention names are written in a single format, *tShay* (Sun) and controlling spelling variations in

NERs ease extracting features and applying it in all words that have found in the same format.

2.8 A nested NEs

It is a word that are formed by a combination of two names, but identifying NE class of a word without contextual consideration leads to wrong classification of NE class and less NER system performance. A word can be classified any of the three NE feature, but when it comes with some words it changes the previous NE class, such kind of problems are common in Amharic. For instance, አዲስ አበባ ዩኒቨርሲቲ Addis Ababa is location name, but when the word ዩኒቨርሲቲ (University) ከተማ አስተዳድር (City Government) and some others are added it becomes Organization names. አዲስ አበባ ዩኒቨርሲቲ (Addis Ababa University) and አዲስ አበባ ከተማ አስተዳድር (City Administration of Addis Ababa). Whereas studying the co-occurred names during model creation overcome challenges that existed in nested NEs.

Chapter three

Related works

There have been different NER systems development attempts taken by different authors throughout the world. In this section an overview of NER will be discussed, by categorizing works on domestic and foreign language. In the domestic languages, one Amharic [6] and two Afan Oromo [8] [7] named entity recognition systems have been developed.

[6] Used CRF to develop a NER system in Amharic with internal and external features, Context features, Part Of Speech tags of tokens, prefix and suffix. The corpus contains 10405 tokens, was split into training, developing and testing data set. To conduct the experiments the researcher used tagged corpuses that are found from news articles, they followed supervised learning method. During the experiment identification of the most relevant features for Amharic NER were attempted and the reported accuracy at first was F-1 73.47% with all of features. When the system was tested by removing POS tag, Prefix and Suffix the performance was 69.70%, 74.61% and70.65%, respectively. The reason,[6] concluded, was that POS tag and Suffix are most relevant to identify Amharic NEs. Even though their research is conducted on an annotated corpus that are taken from WALTA news agency, POS tagging was trained by using Ling pipe tool it is a customizable environment developed for English.

[8] Used a supervised machine learning method and CRF model to develop a NER system in the Oromo language. The features used are both internal and external, normalized tokens, part-of-speech tags, word-shape features, position features, and token prefixes and suffixes. Adopted from CoNLL(Conference on Computational Natural Language Learning), four types of Oromo language NEs are identified: Person, Location, Organization and Miscellaneous. As they

described due to compatibility issue with previously proposed work for Oromo language POS tagging which is external feature in their work they have used ling pipe tool that is originally developed to English. The processing was classified in to pre-processing, training and recognition phase. They have reported that to all processing they have used news based Afan oromo corpus which consists of 23,000 tokens from "*Bariisaa*" and "*Kallacha Oromiyaa"* news papers. After generating the model they reported that on the first experiment the performance obtained are 77.41% Recall, 75.80% Precision and 76.60% F1-measure. They have conducted their experiments by adding the corpus size and reducing features. They found that recall and F1- measure increase while precision reduced while removing feature in the experiment has shown an inverse change in all performance measurements. Finally [8]reported that features play a vital role than the change in the training data size.

The other attempt in the Oromo language is the development of Afaan Oromo Named Entity Recognition (AONER) system. As they reported, AONER used the same method and model that have been used by [8]. The corpus in [7] is also news articles that is collected from Oromia Radio and Television organization, it contains 30,402 tokens. Features used to identify the tokens in to Organization, Person and Location formats are current word, N-grams, previous word, next word, maximum n-gram length, current word shape and surrounding word shape sequence features. The data is split in to training and test set to conduct the experiment. [7] conducted three experiments those are based on current word, the use of n-gram(left/right word) and word shape sequence to identify and detect the boundaries of named entities. As they reported, the overall performance achieved by using the above listed features is F1- measure of 84.71 on test data set. The identified features that have the greatest significance during the experiment are categorized regarding their indication of named entity type. For Named entity type they found that, contextual features (previous, next word, two word previous, and two words next to current word) are a good indicators, clue words are good in identifying all named entity types due to the

reason that has the best predictive value for classifying the NEs and finally as [8]concludes they have also summarized their work, combinations of context and word shape sequence are very important for Afaan Oromo NER.

[3]Used first-order Markov HM-SVM with linear kernel and Expectation Maximization(EM) on the Arabic NER with different feature sets, Tag of other occurrences (GLB), Previous Class (C-1), Trigger(TRG),list of unambiguous names(UNQ),list of most frequent non-person bigrams(BIAMB),effective preceding unigrams(UNIG), Most frequent non-person tokens(AMB), POS tagging (POS), Gazetteers (GAZ) and Lexical (LEX). The data used was ACE corpus is in the form of raw text files of news stories with stand-off annotation in XML files , from the corpus they annotated 70,000 words of the Arabic Tree Bank with their named entity classes. As they reported, ten experiments have been taken place by interchanging the above listed features. they have tested the corpus with two algorithms they have selected, due to the use of different features in the two learning algorithms they did not want to compare the two algorithms but the result found to HM-SVM and EM are 74.49and 69.18, respectively. As [3]reported, best average performance was on the location class, as it was easier in terms of the context of POS tags and also due to the repetition of location entities in the corpus and worst result was on the organization class, since there was no gazetteer employed and also due to the syntactic structure of their naming systems that make them look like any other Arabic phrase. After repeated experiment the identified features those have a positive impact to all NEs are the class of other occurrences of the word being processed together with a list of triggers, class of the previous word and POS

[40] Used a hybrid (Gazetteers and Hidden Markove) model on Indian Languages. They identified four named entities; Person (PER), location (LOC), temple, River and rest are assigning other tag. During the experiment they have used limited sentences that are domain specific (Tourism) corpus. First their experiment tested the corpus by each model independently and scored 40.13%

and 97.3% for Gazetteers and Hidden Markove model, respectively. Hence [19] described that the performance is varied but the corpus size they have used for the two experiments is different, 100 (Gazzetteers) and 40 (Markove) sentences. Finally they concluded that when they applied the hybrid model it scored 98.375%, with this findings [19] reported, that this NER system with high accuracy is built, then this will give way to NER in all the Indian Languages and further an efficient language independent based approach can be used to perform NER on a single system for all the Indian Languages.

[2] Used supervised method with both annotated and unannotated data that was manually tagged at the University of Antwerp. sixteen systems are participated in CoNLL-2003 shared task , Maximum Entropy model is the most used model by a participants and Conditional Random Fields also used by one of the participants , the participants were used different models separately and with combination one with the other. Four types of named entities are tried to detect, persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. The CoNLL-2003 named entity data consists of eight files covering two languages: English and German. For each of the languages there is a training file, a development file, a test file and a large file with unannotated data The English data was taken from the Reuters Corpus and The portion of data that was used for Germen data, was extracted from the German newspaper Frankfurter Rundshau. Main features used by the sixteen systems that participated in the CoNLL-2003 shared task sorted by performance on the English test data. affix information (n-grams, bag of words, global case information, chunk tags, global document information, gazetteers, lexical features, orthographic information, orthographic patterns (like Aa0), part-of-speech tags, previously predicted NE tags, flag signing that the word is between quotes and trigger words. Finally they concluded that, the best performance for both languages has been obtained by a combined learning system that used Maximum Entropy Models, transformation-based learning, Hidden Markov Models as well as robust risk minimization

Conclusion

Literatures that we have reviewed have shown that all of them have followed supervised learning method and statistical approaches, but they have used different features. It clearly shown in [8] and [7] that both of them used the same model and similar corpus with different features from which they have got different performance. From [2] we realized that applications of machine learning models interchangeably for the same or different language would not bring a significant change. Whereas an effective selection, identification and understanding of features and nature of languages are the most important points to come up with a high performance NER system.

CHAPTER FOUR

Design and implementation of the system

4.1 Design of ANER

In this section we have discussed the overall design of our system, Amharic Named Entity Recognizer (ANER). The first part of this section discusses about our data sets which is used for training and testing phases. The second section contains approach we have followed in the study. Then the general overview of the proposed system architecture from the perspective of the system's flow of operations in the form of processes is discussed. Finally, detailed explanation of phases along with subcomponents included in each phase and our system performance measurement are presented.

4.1.1 Data sets

As it has been defined in [6],[7]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. In this ANER, we have used transliterated 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. But we have chosen to use the transliterated one, as we have discussed in section 2.7.4.3, using transliterated corpus reduced spelling variations which are common in Amharic script (fidel) writing. 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. The corpus 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”

and this makes direct use of the corpus difficult. So, removing the string has taken place before going to the main preprocessing phase.

4.1.2 Corpus preparation for ANER

In our system we need two types of data, training and testing data. Training data are used to train the machine learning component. Then after, conducting training and creating a Model, the system has tested on testing data to evaluate the performance of the system. Before going to prepare our data we have been tried to get previously annotated data by [6]; both the researchers and the respective department were not able to provide us the needed data, after several attempt finally we gone to prepare our own data.

By using simple random sampling technique, 13538 tokens from WIC corpus were selected for both phases. During sample selection sentences with a minimum of one NE have been selected, this is with the assumption of if there is no NE in a training data, finding the pattern and generating a good model will be tough and testing such kind of model may have degrade the performance of a system.

After selecting the corpus and conducting preprocessing tasks, removing manually tagged POS and tagging with named entities based on Stanford notation were conducted. According to the format every word in a sentence will be kept on one line together with the NE tag separated by a tab space.

4.1.2.1 Data format

yemaIkelu O
astebabari O
ato O
belayneh PERSON
ayalEw PERSON
lewalta ORGANIZATION
InformExn ORGANIZATION
maIkel ORGANIZATION
IndegeleSut O
maIkelu O
" O
abobako O
" O
yemil O
syamE O
yeteseTewn O
yebeqolo O
zerna O
" O
1xi O
107 O
" O
yeteseNewn O
yemaxla O
zer O
yageNew O
balefew O
and O
amet O
bakahEdew O
mrmr O
new O
:: O

The data contains entity of three types: persons (PERSON), organizations (ORGANIZATION), and locations (LOCATION). Words tagged with O (Others) are words which do not belong to the above three named entities.

4.1.3 Approach Used

From the related works that we have reviewed [22], supervised learning is the current dominant technique for addressing the NER and ANER is designed by following this approach. The machine learning component learns from annotated training data. Conditional Random Fields (CRF) algorithm is the machine learning component we have used. A CRF algorithm has the efficiency of dealing with non-independent, diverse and overlapping features. It has also the capability of overcoming the label bias problem. These two factors make CRF algorithm better suited for NER. The preprocessed annotated data is given to the training phase to create a model, then, testing phase, which is the recognition of the named entities using the model that is generated by the training phase are performed. Figure 4.1 describes our architecture.

4.1.4 Architecture of ANER

Under this section, the architecture of the system will be presented. Figure 4.1 shows the overall functionality of the Amharic Named Entity Recognition system. There are three main phases in our ANER system, Preprocessing, training, and testing. These three major phases might be used in different NER and other machine learning systems, but tasks which are taken place within each phase are different and this is caused by the data we used and the procedure that a researcher has chosen. Because of the above reasons, there is no single architecture that we all should follow. With this assumption our ANER architecture is described as follows.

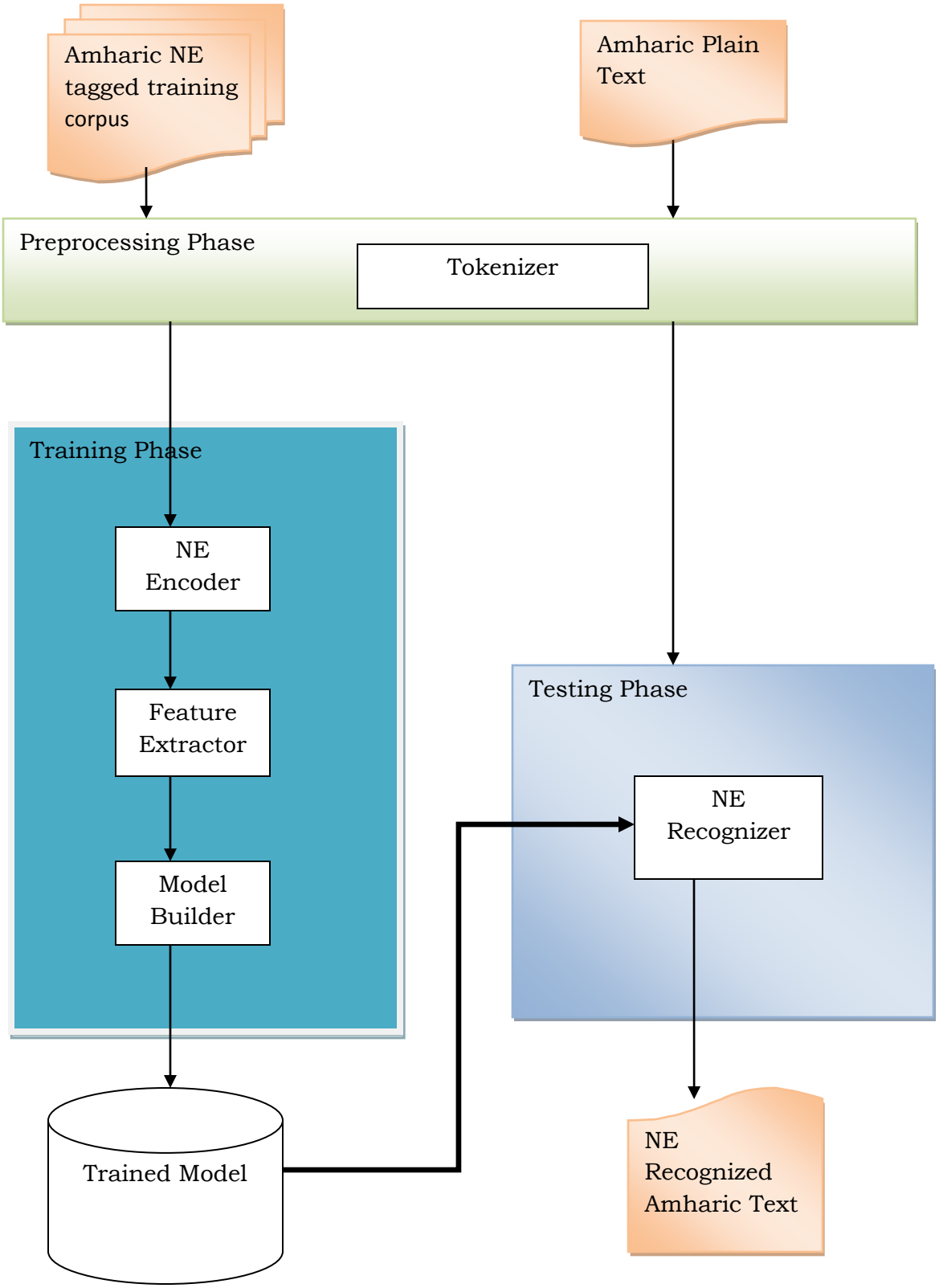

Figure 4.1 Architecture of ANER.

4.1.4.1 Processes of ANER

ANER is designed in a manner that, first it learns properties and parameters associated with NEs from the training data. It then receives input plain text and predicts the possible NEs out of the plain text. The architecture has two processes: learning process and prediction process.

4.1.4.2 Learning process

Training is handled by components in the learning process. Preprocessing is initially performed on the training corpus. The corpus is tagged based on Stanford NER format. Then it is passed to the NE encoder which identifies a token and its corresponding NE tag through encoding. The token and tag sequence generated by the NE encoder is handled to the Feature Extractor. Feature Extractor extracts necessary features to identify NEs based on the generated token and tag sequence, the extracted features will then be used as an input to the Model Builder. After the above all fulfilled, the builder begins the model building procedure to generate a trained model. Generally, here in the learning process, based on the training corpus we have generated a model that will be used to predict NE classes of testing.

Pseudo code for learning process:

For *all tokens in the training data*

Activate *previous word sequence feature*

Based on the specified maximum length of left and right window size

fExtractor .extract(previous word feature for the current token and set as 1)

Activate *next word sequence feature*

Based on the specified maximum length of left and right window size

fExtractor .extract(next word feature for the current token and set as 1)

Activate *word pair feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(word pair feature for the current token and set as 1).

***Activate** word shape feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(word shape feature for the token and set as 1).

***Activate** NE tag feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(NE tag feature for the current token and set as 1).

fExtractor.append(the NE tag of the previous token and set as 1).

fExtractor.extract(the NE tag of the next token and set as 1).

***Activate** prefix and Suffix feature*

Based on the specified maximum length of prefixes and suffixes

fExtractor.extract (prefix & suffix feature for the current token and set as 1).

End For

Return fExtractor().

4.1.4.3 Prediction process

Here the plain text is preprocessed. Tokenization is the only preprocessing task performed on the input plain text. The tokenized data is then passed to the NE Recognizer. The trained model will assist the NE Recognizer to perform such tasks as calling stored features and NE tagging on the input data. The NE Recognizer recognizes NEs out of input text with the aid of the trained model. The result, NE recognized Amharic text is then supplied as an output.

4.1.4.4 Preprocessing

4.1.4.4.1 Tokenization

Tokenization can basically be defined as the process of breaking up a text into its part of elements, tokens. In Amharic, tokenization process is easier since every word is separated by a space. Tokenization is done on the input plain text during the prediction process. Tokenization on the training corpus is also done during corpus development. The tokenization takes the input text supplied from a user and tokenizes it into a sequence of tokens that can make it easy to recognize NEs.

4.1.4.5 Training phase

Training phase contains necessary components that are used in the process of training the machine learning component. The components within this phase operate on the data from the training corpus and they are the NE Encoder, Feature Extractor and the Model Builder.

4.1.4.5.1 NE encoder

This is a process that provides information to feature extractor by identifying a token and its tag. From the training data the encoder analyzes which one is a token and which one is a tag then supplies to the next process. Hence, the encoder provides important clue to feature extractor, setting it wrongly also leads to wrong model creation. To avoid this we have kept the consistency between corpus and NE encoder.

4.1.4.5.2 Feature extractor

As we have discussed in chapter two, Features are properties of a text that are used to provide necessary information associated to a given NE and increase the confidence level of predicting a token as a NE. ANER's feature extractor is responsible for identifying and extracting all the necessary features from the training data. It is one of the essential components that supply necessary information (features) during the building of the model. The feature extractor is designed to extract features from the training data and store them for future use.

The feature extractor that we have used, CRF algorithm, performs repeated feature extraction from the same context for different positions in the input. All the extracted features are supplied to the model builder which will predict the parameters of the model. To identify Amharic NEs we have used: Word Prefix and suffix, word shape features, word position features with a different window sizes and the detailed feature combination used are described in section 4.2. As we have discussed in chapter two POS tag feature is used in different NER systems and they have scored various performances, but with regard to the tool we have chosen (Stanford), the features used in our work are very similar to POS tags. Due to this reason we have focused on using other features that predict NEs of Amharic.

4.1.4.5.3 Model Builder

The main concern with the machine learning component is to build a trained model that will be used in prediction. Building a model is the component designed to estimate the model Coefficients and then build a trained model. Estimation is taken place based on the input from the training corpus that is supplied by extracted features which is generated from feature extractor. The model builder is designed to perform all the calculations we have discussed in section 2.5 and generated a trained model that will be used in prediction phase.

4.1.4.6 Prediction phase

This prediction or recognition is the final phase in NER system. It is a process of recognizing NEs from the given preprocessed token. There are two sub tasks are taken place in this phase, detection and classification.

4.1.4.6.1 Detection

Detection is the first task which finds candidate tokens that can be NEs from the tokenized plain text. By using the knowledge acquired from the model, detection is performed. The knowledge in model builder contain features that are extracted and stored during the training phase, are supplied to the recognizer to identify NEs from the given plain text. Identification of NEs are performed based on the calculated probability.

4.1.4.6 .2 Classification

After detection of NEs for the candidate tokens are performed, classification process is preceded by tagging the detected candidate tokens. During classification each token in the given plain text will be tagged with their possible NE tags using Stanford scheme. Finally, the recognizer will generate NE recognized Amharic text and send it as an output.

4.1.5 Performance evaluation

As we have discussed under Corpus preparation for ANER, for both phases (training and testing) 13,538 tokens were used. For training and testing we have

used, 12296 and 1242, respectively. 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).

$$P=\frac{T_P}{F_P+T_P}$$

And

$$R=\frac{T_P}{F_N+T_P}$$

Where, True Positives (T_P) are NE tags which are produced by the model that match the reference tagging , False Positives(F_P) are NE tags returned by the model that are not in the reference tagging.
And lastly, False Negatives (F_N) are NE tags in the reference tagging bur they are missed by the model.

The F measure which is computed by the uniformly weighted harmonic mean of precision and recall:

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

Performances that we have discussed in the following section are performed according to these performance measurements.

4.2 Experimentation Result

In this section we have discussed experiments which are conducted in different scenarios. The different scenarios have taken by varying our feature sets. The first section presents corpus used in the experiment then describing feature set in the experiment. In the second section experimentations with their performance achievement are presented. The final section discussed experiment results with their graphical representation.

4.2.1 Corpus size

The number of tokens used in this work are 13538, from these 10% of the total tokens are used for testing and the rest are supplied for model creating.

Table 4.1 corpus size

Number of tokens in training phase	Number of tokens in testing phase	Total number of tokens
12296	1242	13538

Feature sets plays a crucial role in CRF frameworks, and as we have stated under objective of the study, finding the optimal feature sets for Amharic language is our intended objective. To find these feature sets we have used word based features contains suffixes and prefixes of the current and surrounding tokens and context based features that are previous and next words of a particular word with their named entity tags.

- Suffixes- 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 four of the current and surrounding word have been taken for this language since it gave better performance than using length up to 2.

- Prefixes - A fixed length prefix of the current and the surrounding words might be treated as features, in this work, the tool we have used did not allow us to use prefix and suffixes separately.
- Previous word- words that occurred before to a current word and it might be also a clue word that indicates NE class of succeeding word.
- Next word- words that occurred next to a current word and it might be also a clue word that indicates NE class of preceding word.
- Named Entity Tags- the NE tags are used in our work as a feature, these features are used by increasing our window size up to 4.
- Orthographic – word shape of a token was one of our features set. The experiment also conducted by increasing the window size for orthographic feature and combining it with other features.
- Word pairs - features that are taken by calculating frequency co-occurrence of words.

The above listed features are used in this work, but before deciding feature set of this system determining the first feature set for our baseline experiment was the toughest task during experimentation. This raised due to a reason that the earlier ANER system feature set are not compatible with the tool we have chosen, Lingpipe which is a tool that has been used by previous ANER researchers. It accepts suffix and prefix independently, beginning of sentence (BOS) and End of a sentence (EOS) are also additional features in their tool. The other feature is Part Of Speech tagging (POS), the feature that we have not used. As we have discussed previously, Stanford features have a capability that POS can performs. According to Stanford NER developers group, the features used by POS tagger are very similar to those used in the NER system, so there is very little benefit to using POS tags[3].

With all the above mentioned factors, using previous work features as a baseline experiment were impossible. Hence, starting our experiment with all

[3] http://www-nlp.stanford.edu/software/crf-faq.shtml#pos

the same features that have been used in the previous work is difficult we have selected common features from them. But feature usages are still different.

For our baseline experiment we have selected the following features that have been used and brought the optimal performance in previous work.

Table 4.2 Baseline experiment

Experiments	Features used	Performance		
		Recall %	Precision%	F-measure %
Baseline	Window size (left and right)=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag	76.84	82.95	79.78

NB. The features that we removed from previous work are BOS, EOS and POS, performance achieved by features in the table was F-measure of 74.61 % and it was the highest performance in their work. This performance was achieved with 10405 total token sizes.

After we have selected baseline experiment features for our work, identifying optimal feature set for ANER is continued. In order to begin the first experiment, we have started by selecting a single feature with a single window size; this is because of the assumption that studying a role of each feature independently would help to come up with optimal feature sets. With this reason previous word feature with one window size has chosen for the first experiment.

Table 4.3 Experiment one

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
1.	Left Window size=1 Previous word	52.6	75.94	62.15

The second experiment is conducted by increasing the window size of previous experiment by one that is two. Window size increments added was on the left side only and F-measure increased by 0.20%.

Table 4.4 Experiment two

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
2	Left Window size=2 Previous word	52.6	76.51	62.35

From table 4.4 we understood that window size increment has performed better than a single window size so, the next experiment used bigger window size. The third experiment was conducted by taking next word along with the above window size. The F- measure is the same as experiment 2 (i.e. previous word with window size two).

Table 4.5 Experiment three

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
3	Left Window size=2 Next word	52.6	76.51	62.35

After analyzing the influence of next and previous word feature independently the next step was studying the role of these two features combination. Combination of two features is taken with a window size of two to the left side.

Table 4.6 Experiment four

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
4	Left window size=2 Previous word Next word	52.33	75.94	61.97

As it is shown in Table 4.6 F- measure is reduced by 0.38% from F-measures achieved when features applied independently. Even though the F- measure is reduced, using previous and next word in combination has its own advantage. F-measure was not the only decreasing performance recorded but also Recall and precision too. Based on experiment 4 additional features are selected. Now, the identified features which positively recognized NEs are two window sizes to the left, previous and next words. The next experiment was conducted by using suffix and prefix with a length of two and four. As we have discussed in section 2.7.4.3**,** this work used transliterated news and most Amharic scripts are represented by two English alphabets. With this assumptions our instant prefix and suffix length started from two and finally reached to four.

Table 4.7 Experiment 5 and 6

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
5	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 2	63.87	75.31	69.12
6	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4	65.46	76.05	70.36

In table 4.7 the F- measures of our system has increased on both experiments, experiment 5 and 6 by 6.77 and 8.21, respectively. Then, based on the above result we have preceded to the next experiment by adding previous and next NE tags independently and combination of them to experiment 6. From our previous experiment (i.e. experiment 6) results in experiment 7, 8 and 9 shows, F- measure has increased by 9.42, 7.69 and 9.97, respectively.

Table 4.8 Experiment 7, 8 and 9

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
7	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag	76.84	82.95	79.78
8	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Next word tag	76.19	80	78.05
9	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag	75.52	85.8	80.33

Here in experiment 9, the F- measure the system scored increases the baseline experiment by 0.55%. The next experiment has taken all the features in experiment 9 with a new window size. In experiment 10 we took window sizes of two, 1 to the left and right side of the current word.

Table 4.9 Experiment 10

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
10	Window size (left and right)=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 2 Previous word tag Next word tag	76.84	84.88	80.66

The F-measure of experiment 10 is increased by 0.33 % from our previous experiment result. The following experiment is conducted to show the role of other additional features. First we took all features from experiment 9 then, add orthographical and word pair features independently, finally combination of these two features has also been tested as follows:

Table 4.10 Experiment 11, 12 and 13

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
11	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape	75.52	85.8	80.33

12	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word Pairs	74.48	86.67	80.12
13	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape Word pairs	74.1	86.67	79.9

The above three experiments have shown that , orthographical features brought the same F- measure , Precision and Recall as features which have been used in experiment 9. Whereas , experiment 12 which is addition of word pairs feature only and experiment 13 , combination of word pair and orthographical feature have scored less F- measure performances than the previous experiment. During experiment 12 and 13 the F- measure recorded was reduced by 0.21% and 0.43 %, respectively.

The final experiment that we have conducted has taken all features in experiment 13 with one additional window size to the left. Here in experiment 14 F- measure has increased by 0.1% from experiment 13.

Table 4.11 Experiment 14

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
14	Left Window size= 2 Right window size=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape Word pair	76.84	83.43	80

4.2.2 Discussion of Experiment results

The above experiments show that different combination of feature sets have brought various F- measures, Recall and Precision. The following table presents performance of the systems.

Table 4.12 Performance

Experiments	Performance		
	Recall %	Precision%	F-measure %
Baseline	76.84	82.95	79.78
1	52.6	75.94	62.15
2	52.6	76.51	62.35
3	52.6	76.51	62.35
4	52.33	75.94	61.97
5	63.87	75.31	69.12
6	65.46	76.05	70.36
7	76.84	82.95	79.78
8	76.19	80	78.05
9	75.52	85.8	80.33
10	76.84	84.88	**80.66**
11	75.52	85.8	80.33

12	74.48	86.67	80.12
13	74.1	86.67	79.9
14	76.84	83.43	80

From baseline to the last experiment that is experiment 14, removing and increasing the number of features sets and window size have been taken placed. Removing and adding of features are followed researchers best rationality.

Features we have used in the baseline experiment are selected by the method that we have mentioned above.

To select the first experiment features we started from the baseline experiment and study the role of each feature in ANER. With this assumption analyzing the result of previous and next word features are conducted by changing the window sizes.

By using of previous and next words independently with the same window size we have achieved the same precision, recall and F- measure. After studying the performance from experiment 1, 2 and 3, the features identified for the next level were previous word, next word and window size of two from the left side of a current word. The main finding here is, a different window size have changed the performance in each steps and this clearly shows previous and next word features independently with a window size of two have performed better than a single window size.

Based on this understanding experiment 4 is conducted and it achieved less performance. This experiment shows, a combination of previous and next word with different window sizes degrades performance of ANER system. After conducting different combination of the above three feature sets, looking for the next feature set is continued by using all features that have been used in experiment 4.

In experiment 5 and 6, the length of prefix and suffix we took shows a great change from previous experiments. By varying lengths of these two features are also indicated there is a change in the performance. From these two experiments we have observed that prefix and suffix could be one of the optimal

features in ANER system. Then, by using features in experiment 6 we continued for further optimal features and studying the role of previous and next word tags of a current token were taken placed.
Here in the following experiments we have followed the same procedure as previously conducted experiments. From the three experiments that take each feature independently and their combination shows a tremendous change from experiment 6. As we have observed in the experiment, the combination of previous and next tag performs better than performances that were recorded when they are added independently. But previous tag performs better than next word tag.

After experiment 9, we have gone back to the baseline experiment, this is because of in experiment 9 combination of previous and next tag outperforms on applying previous tag only. By taking the result of experiment 9 into consideration, experiment 10 was conducted by adding next tag to features in the baseline experiment and we got the highest performance of the entire experiments.

Finally , based on the performance achieved in experiment 9 finding other optimal features are supported by adding orthographical feature, word pair feature and combination of them with different window size were conducted. The result from the experiment shows, a simple addition of many features could not improved the performance of ANER system.

The general graphical representation of the above experimentation discussion is presented by clustering our fifteen experiments in to five clusters. Clustering is based on feature set which have been used in the experiments. The first cluster contains experiment 1, 2, 3 and 4; those are conducted by using previous and next word features with different window sizes. Experiment 5 and 6 are grouped in to second scenario that clearly shows performance changes over prefix and suffix features with different length. Previous and next word tag used in

experiment 7, 8 and 9 are clustered in to the third scenario. The fourth cluster (experiment 11, 12, 13 and 14), holds word pair and word shape feature sets with different window sizes. Finally, baseline experiment and the highest performance experiment (i.e. 10) are fallen in to cluster 5.

Based on the above clustering , first the change within each clusters are presented then the second graphical representation held five experiment results by taking transition performance, which is an experiment performance that we added a feature to precede for the next experiment. In the second type of graphical representation the researchers took experiment 14 from the fourth cluster, this is because there was no transition after this experiment and this is one of the experiment that used all feature set of ANER system.

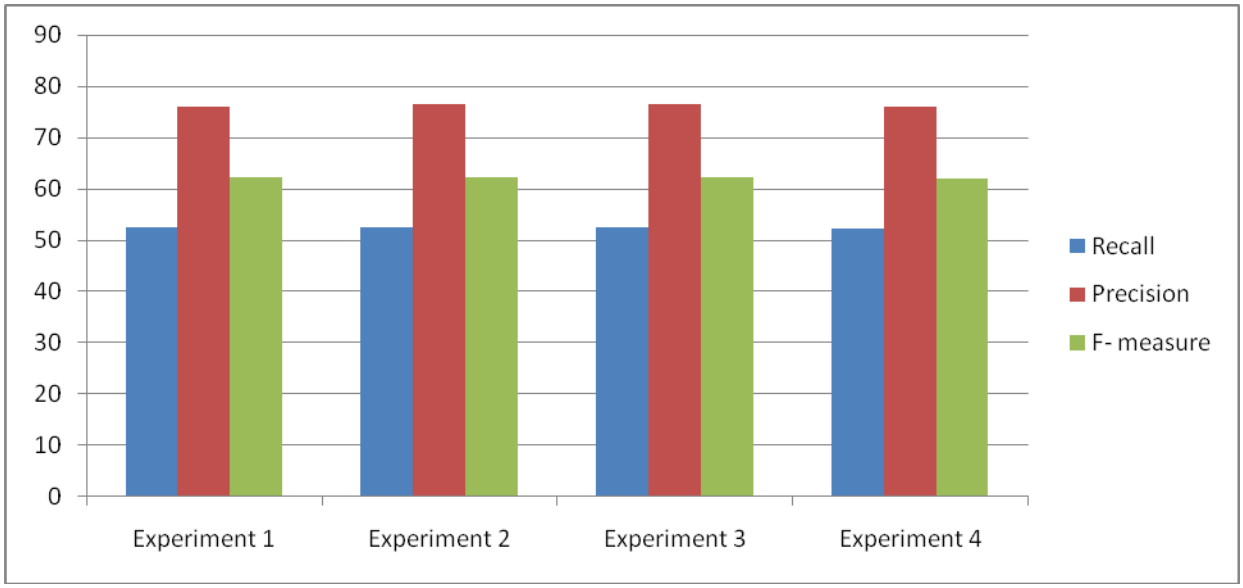

Figure 4.2 Scenario 1.

The first scenario shows F- measure did not increase when we change window sizes of previous and next word features .

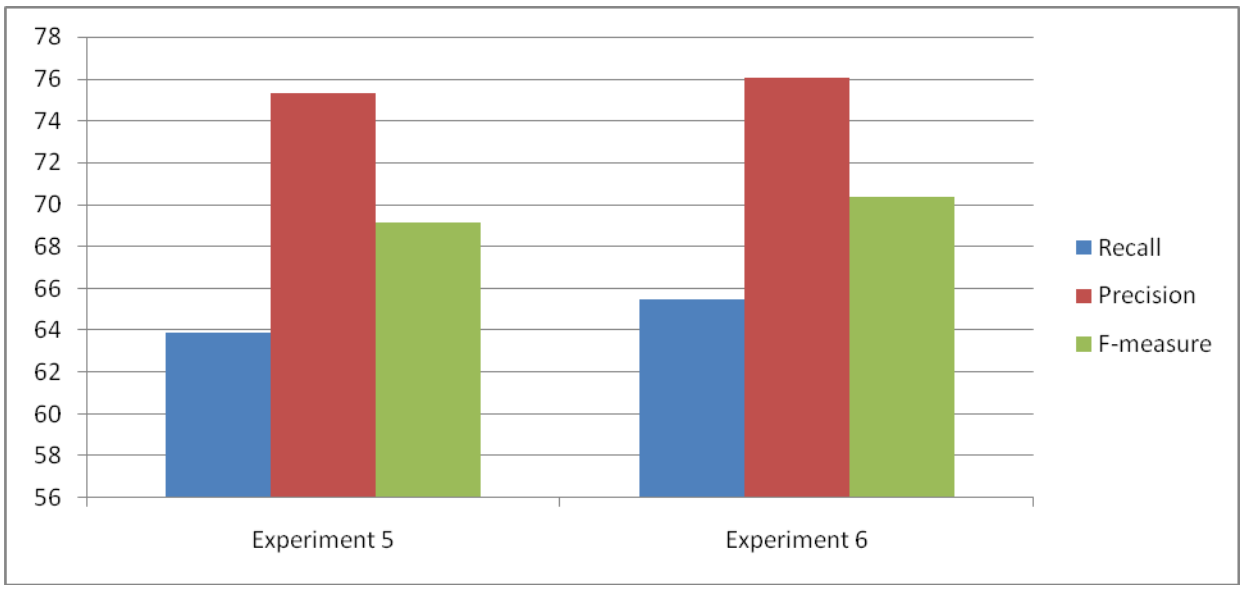

Figure 4.3 Scenario 2

It is our first observation that prefix and suffix have played a role in identifying Amharic NEs. As it is shown in the above graph within each experiments suffix and prefix length also alter NE recognizing ability of a system.

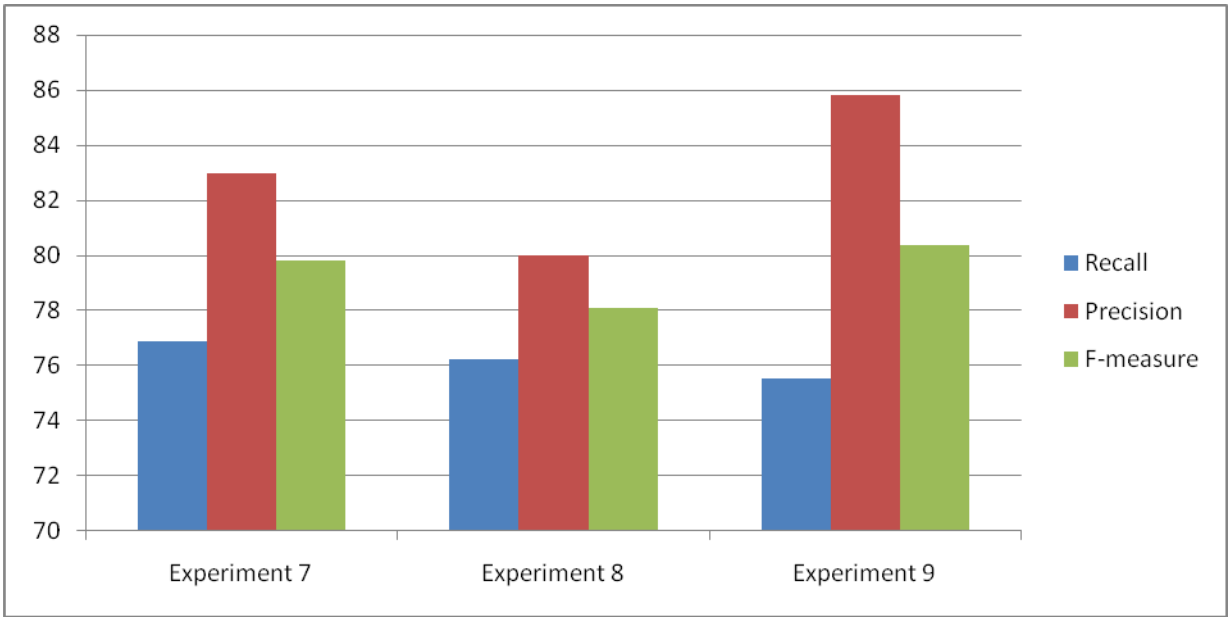

Figure 4.4 Scenario 3

After prefix and suffix the drastic change has been observed during previous and next tag features. The change in each feature has changed; The worst performance during scenario 3 is independent usage of next word feature and the maximum was combination of the two features. The precision in this scenario is one of the greatest from previous experiments.

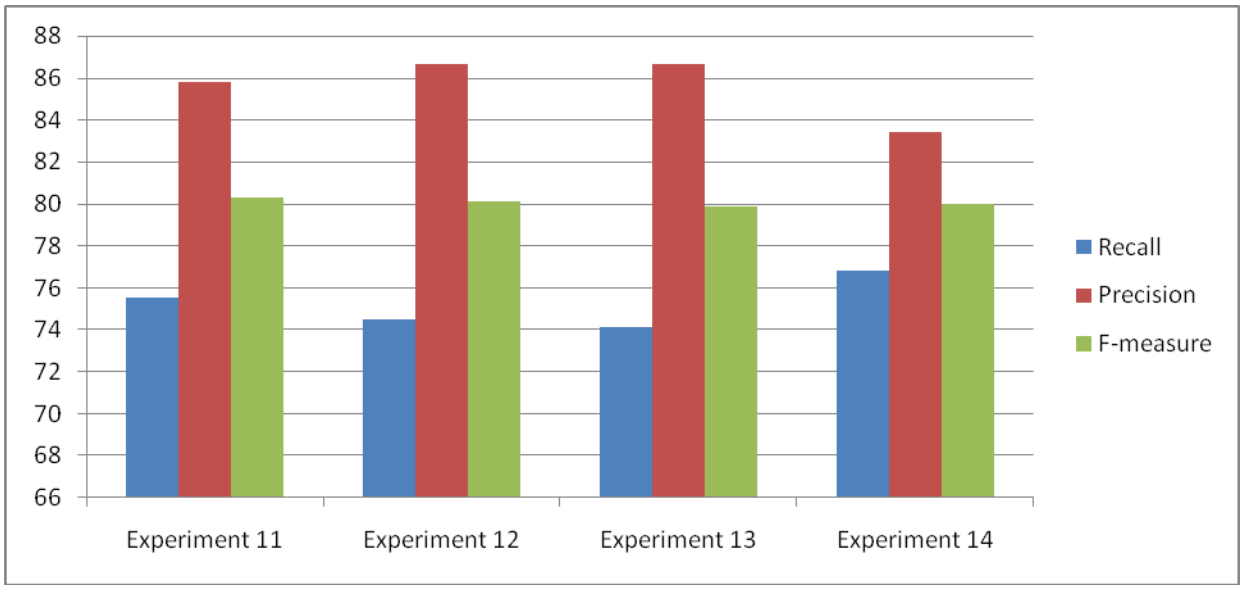

Figure 4.5 Scenario 4

As compared to scenarion3 the change in scenario 4 is insignificant . Recall and precision in scenario 4 are varied but F- measure in experiment 11 repeats experiment 9 result, where as the rest experiments performed with less F-measure that experiment 11 scored. The performance in scenario 4 indicated after certain increment of feature set the performace will saturate and with worst case it inclined to reducing F- measure of the system . The reason behind this performance reduction need further study.

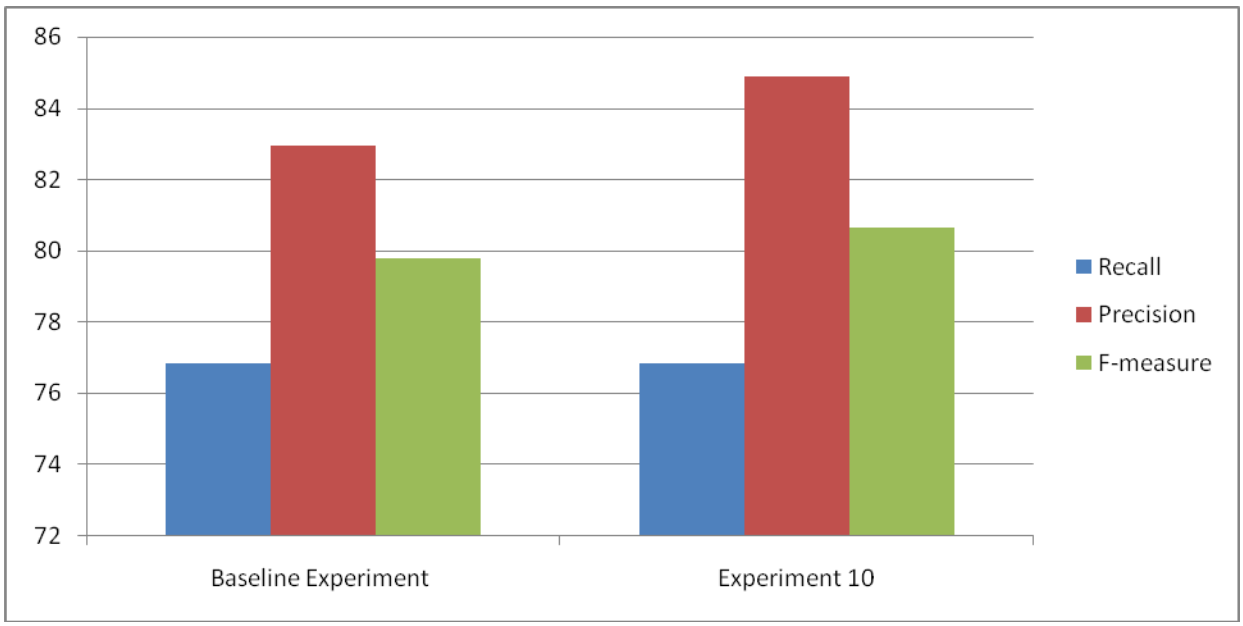

Figure 4.6 Scenario 5

The final cluster in the first graphical representation is scenario 5. In this scenario precision, recall and F- measure changes from baseline to experiment 10. As it is mentioned in scenario 3 combinational usage of previous and next tags have a potential to recognize Amharic NEs and the new finding in this scenario is , performace of a system varied in combinational usage of previous and next tags with different window sizes.

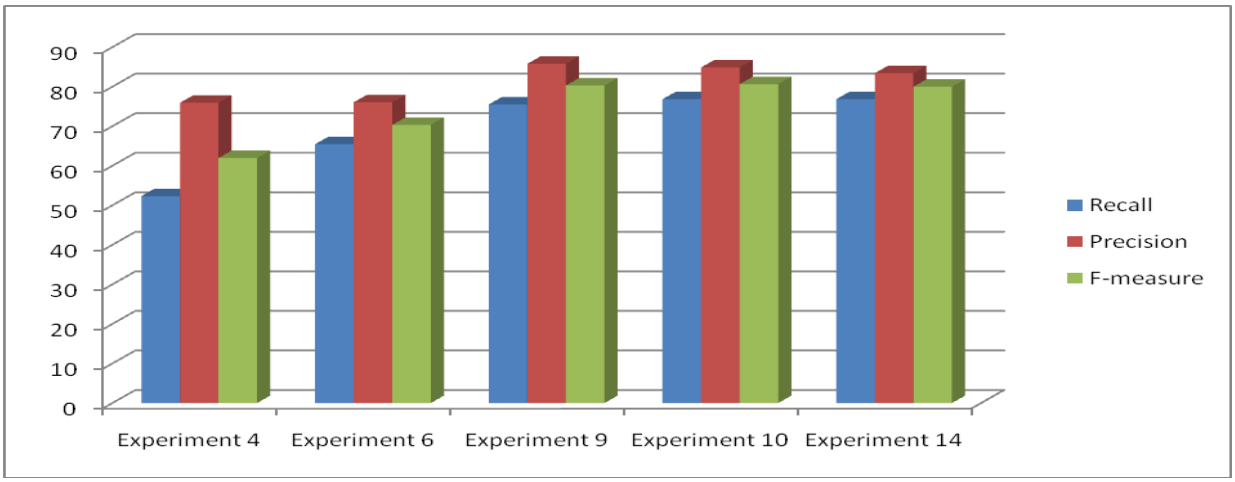

Figure 4.7 Second scenario graphical representation

From the above graph we understood that, Recall, Precision and F- measure of the system have been changing from experiment 4 to 6 and 9 in a tremendous way then it increased with a little decimal value in experiment 10. Finally the performance of the system reduced in experiment 14.

Generally, during our experimentation the identified optimal features that bring tremendous change are : suffix and prefix with 4 word length , previous and next word tag of a current word and window size of features.

Chapter five

Conclusion and recommendation

5.1 Conclusion

Named entity recognition is the process of identifying the right tag for names in a document. Many NER systems are developed for English and other European languages, next to these, NER for Arabic and south Asian languages have been developing. Amharic is one of the languages that have many speakers in Ethiopia, but it is under resourced.

This work tried to contribute one important point which plays a role for overcoming some challenges in NLP regarding to Amharic. Having a good NER system could be used in information extraction of any domain specific tasks, question answering, information retrieval … etc. to come up with a system that could be used in the above application areas, identifying optimal feature set is the most important task of any NER. This work is also delimited to identifying these features to recognize PERSON, ORGANIZATION and LOCATION names of transliterated Amharic corpus.

This study has proposed and designed a system called Amharic Named Entity Recognition. The system is designed based on a supervised machine learning component. We implemented the machine learning component using CRFs algorithm. To do this research, 13538 tokens were taken from WIC corpus and tagged manually. These data sets are classified into training and testing. By using the data processes of identifying optimal features for ANER have been implemented, we have conducted fifteen experiments and the roles of each feature in different experiments were tested. The highest performance achieved in this work is 80.66%, with a window size of two on both sides, previous and next tag of a current word and prefix and suffix with length four. The worst performance achieved is 61.97%, feature sets are a window size of two from the left side of a word and previous and next word of the current token.

The previous work on ANER has performed with 74.61 on a different tool, corpus size and feature set. Due to these reasons we could not compare this work with the previous one. But from the finding we got, a tool might have impact on the performance of ANERs.

From the findings, prefix and suffix with a length of four, previous and next NE tag of a token and window size features are the observed optimal feature sets for recognizing Amharic named entities.
During combining a feature set one needs to be careful in grouping the features; otherwise it degrades system's predicting ability.

5.2 Recommendation

This study showed some of the features that have never got an attention of previous ANER. The involvement of those new features has brought a promising performance and the following points are some of our observation that should be taken in consideration for future works of ANER.

- Due to the limitation of the tool that is used for this study, we could not observe the separate role of prefix and suffix. So, separating these two features and identifying whether their combinations or independent role should be included in ANER optimal feature set is recommended.
- Experimentation with regard to corpus size has never been conducted in this work and the corpus size in this work is very small with regard to CRF based NER systems and training by different corpus size would give a better performance
- From literatures we have reviewed[7, 8] used Lingpipe and Stanford, respectively and performance achieved with different corpus size and features sets were 76.6 and 84.71, respectively , these shows further research regarding to NER tool needs to be.

- Developing a hybrid approach which incorporates a rule based on the nature and pattern of Amharic NEs with CRF based approach is recommended since it might give a better performance.

References

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes,* vol. 30, pp. 3-26, 2007.

[2] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 142-147.

[3] S. M. Algahtani, "Arabic Named Entity Recognition: A Corpus-Based Study," 2012.

[4] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1-7.

[5] A. A. Argaw and L. Asker, "An Amharic stemmer: Reducing words to their citation forms," in *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, 2007, pp. 104-110.

[6] M. A. Mehamed, "Named Entity Recognition For Amharic Language," Msc., Computer Science, Addis Ababa University, Addis Ababa, 2010.

[7] H. Keno, "DESIGNING A NAMED ENTITY RECOGNITION SYSTEM FOR AFAAN OROMO TEXT," MSC, Information Science, Addis Ababa University, Addis Ababa, 2012.

[8] M. L. Kejela, "Named Entity Recognition for Afan Oromo," Computer Science, Addis Ababa University, Addis Ababa, 2010.

[9] S. S. Keerthi and S. Sundararajan, "CRF versus SVM-struct for sequence labeling," Technical report, Yahoo Research2007.

[10] D. Li*, et al.*, "Conditional random fields and support vector machines for disorder named entity recognition in clinical texts," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, 2008, pp. 94-95.

[11] P. Thompson and C. Dozier, "Name searching and information retrieval," in *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, 1997, pp. 134-140.

[12] T. Sehgal*, et al.*, "NLP & its Applications."

[13] A. Mikheev*, et al.*, "Named entity recognition without gazetteers," in *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 1999, pp. 1-8.

[14] E. Riloff and J. Lorenzen, "Extraction-based text categorization: Generating domain-specific role relationships automatically," *Natural language information retrieval,* pp. 167-196, 1999.

[15] J. Cowie and W. Lehnert, "Information extraction," *Communications of the ACM,* vol. 39, pp. 80-91, 1996.

[16] M.-F. Moens, *Information extraction: algorithms and prospects in a retrieval context* vol. 21: Springer, 2006.

[17] K. Kaur and V. Gupta, "Name Entity Recognition for Punjabi Language," *Machine translation,* vol. 2, 2012.

[18] R. Farkas, *et al.*, "The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning---Shared Task*, 2010, pp. 1-12.
[19] A. Kao and S. R. Poteet, *Natural language processing and text mining*: Springer, 2007.
[20] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning,* vol. 3, pp. 1-130, 2009.
[21] O. Chapelle, *et al.*, *Semi-supervised learning* vol. 2: MIT press Cambridge, 2006.
[22] D. Nadeau, "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision," Phd, University of Owtawa, Canada, 2007.
[23] E. Alpaydin, *Introduction to machine learning*: The MIT Press, 2004.
[24] J. Mayfield, *et al.*, "Named entity recognition using hundreds of thousands of features," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 184-187.
[25] M. Tkachenko, *et al.*, "Named entity recognition: Exploring features," in *Proceedings of KONVENS*, 2012, pp. 118-127.
[26] Y. Benajiba, *et al.*, "Arabic named entity recognition using optimized feature sets," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 284-293.
[27] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 2009, pp. 147-155.
[28] A. Ekbal and S. Bandyopadhyay, "Bengali named entity recognition using support vector machine," in *Proceedings of workshop on NER for south and south east Asian languages, 3rd international joint conference on natural languge processing (IJCNLP)*, 2008, pp. 51-58.
[29] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 2003, pp. 134-141.
[30] R. Klinger and K. Tomanek, *Classical probabilistic models and conditional random fields*: TU, Algorithm Engineering, 2007.
[31] J. Lafferty, *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.
[32] B. Sun, "Named entity recognition: Evaluation of Existing Systems," Norwegian University of Science and Technology, 2010.
[33] M. Marrero, "Advances in Computational Linguistics," *Research in Computing Science,* vol. 41,2009, pp. 47-58, 2009.
[34] C. W. Isenberg, *Grammar of The Amharic Language*. London: Richard Watts 1976.
[35] B. M. HAILEMARIAM, "N-gram-Based Automatic Indexing for Amharic Text," Msc, SCHOOL OF INFORMATION STUDIES FOR AFRICA, Addis Ababa University, Addis Ababa, 2002.
[36] G. A. Demeke and M. Getachew, "Manual annotation of Amharic news items with part-of-speech tags and its challenges," *Ethiopian Languages Research Center Working Papers,* vol. 2, pp. 1-16, 2006.
[37] A. A. Argaw, "AUTOMATIC SENTENCE PARSING FOR AMHARIC TEXT AN EXPERIMENT USING PROBABILISTIC CONTEXT FREE GRAMMARS," Msc, Computer Science, Addis Ababa University, Addis Ababa, 2002.

[38] L. Bender and C. Ferguson, "The Ethiopian writing system," *Language in Ethiopia. Edited by ML Bender, JD Bowen, RL Cooper, and CA Ferguson. London: Oxford University press*, 1976.
[39] T. Zhang and D. Johnson, "A robust risk minimization based named entity recognition system," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 204-207.
[40] N. Jahan*, et al.*, "Named Entity Recognition in Indian Languages Using Gazetteer Method and Hidden Markov Model: A Hybrid Approach."

Appendixes

Appendix 1. The Amharic character set [38]

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					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ᎀ	ᎁ	ᎂ	ᎃ	ᎄ	ᎅ	ᎆ

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

A Named Entity Recognition for Amharic

By

Besufikad Alemu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE:

Addis Ababa, Ethiopia

June, 2013

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

A Named Entity Recognition for Amharic

By

Besufikad Alemu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE:

Addis Ababa , Ethiopia

June, 2013

DECLARATION

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

_____

Besufikad Alemu

June, 2013

Thesis has been submitted for examination with our approval as University advisor.

_____

Solomon Teferra (PHD)

June, 20

Acknowledgement

First and foremost, I would like to thank God, who makes everything possible. I would extend my sincere gratitude to my advisor Dr. Solomon Teferra for his critical comments and for being my driving force throughout this work.

My special thanks go to my sister, Yilfashewa (Beti), for making all those hardships I had to go through easier. Beti, you are the greatest sister, mom and best friend. I could not have been able to make it without you.

I would also thank Moges Ahmed and my friends for their understanding and support.

Contents

List of table

List of figure

Abbreviation

MUC :	Message Understanding Conference
IE :	Information Extraction
NERC:	Named Entity Recognition and Classification
ORG:	Organizations name
PER :	Persons name
LOC :	Locations name
CONLL :	Conference on Computational Natural Language Learning
NER :	Named Entity Recognition
POS:	Part of Speech tag
NERs:	Named Entity Recognition System
NLP :	Natural Language Processing
ANER :	Amharic Named Entity Recognition
CRF :	Conditional Random Fields
IE :	Information Extraction
U.S.:	United States
SL :	Supervised Learning
HMM :	Hidden Markov Models
ME :	Maximum Entropy
SVM :	Support Vector Machines
UL :	unsupervised learning
NLTK :	Natural language tool kit
EM :	Expectation Maximization
SSL :	Semi- supervised Learning

Abstract

This thesis describes the development of Named Entity Recognition (NER) system for Amharic. 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. Amharic Named Entity Recognition (ANER) is proposed based on supervised machine learning, Conditional Random Fields (CRF). Amharic corpus of size 13,538 words have been developed with Stanford tagging scheme.

Since the objective of the work is working on feature set of Amharic Named Entities (NE), fifteen experiments were conducted by interchanging different features. Previous and next word, named entity tag of a word, word pairs, word shape, prefix and suffix features have been used to identify three NEs, Person, Organization and Location. The results of the experiment show that proper features combination in NER has a great role. The highest F-measure achieved in this work is 80.66%, with a window size of two on both(left and right)sides, previous and next tag of a current word and prefix and suffix with length four. The worst performance achieved is 61.97%, feature sets are a window size of two from the left side of a word and previous and next word of the current token. Finally the identified optimal feature sets from the experiment are: prefix and suffix with a length of four, previous and next NE tag of a token and window size features

Key word: Named Entity Recognition, Amharic Named Entity Recognition, Conditional Random Fields, Named Entities

Chapter One

Introduction

1.1. Background

According to (Grishman and Sundheim ,1996) as they are cited on [1], the term "Named Entity", now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC-6). At that time, MUC was focusing on Information Extraction (IE) tasks where structured information of company activities and defense related activities is extracted from unstructured text, such as newspaper articles. In defining the task, people noticed that it is essential to recognize information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions. Identifying references to these entities in text was recognized as one of the important sub-tasks of IE and was called "Named Entity Recognition and Classification (NERC)" [1].

Named entities are phrases that contain the names of persons, organizations and locations. Example:

[ORG AAU] Official [PER Besufikad] heads for [LOC Addis Ababa] .

This sentence contains three named entities: Besufikad is a person, AAU is an organization and Addis Ababa is a location. Named entity recognition is an important task of information extraction systems. There has been a lot of work on named entity recognition, especially for English [2].

According to [1], a good proportion of work in NER research is devoted to the study of English but a possibly larger proportion addresses language independence and multilingualism problems. German is well studied in CONLL-2003 and in earlier works. Similarly, Spanish and Dutch are strongly represented, boosted by a major devoted conference: CONLL-2002. Japanese has been studied in the MUC-6 conference, in addition to the above mentioned

languages he has listed many European languages that have been tried to develop a good NER.

1.2. The challenge of Named Entity Recognition

The ambiguity of NEs in text resides at two different levels; detection and classification. The first involves disambiguating NEs from non NEs in text and the second involves classifying them into classes e.g. person or location. The two processes of NER are sequential and the correctness of the identification phase is a prerequisite for the classification phase. The second level is dependent on the first one; NEs will not be classified correctly if they are not detected correctly in the first place [3].

The level of ambiguity that resides at these two levels differs from one language or domain to another. In English text, for instance, the identification would normally rely on orthographic case information (lower and upper) to detect NEs. If we manage to identify all NEs using that information then the fundamental problem is classifying them. For the classification, assuming that we managed to collect and group all NEs into lists for each NE class, there still would be cases where one token might fall into more than one list [3]. For instance, in Amharic the word *Amede* could be a person name or location. According to [3], Thus more information is required to classify NEs correctly; for example, the context in which the word occurs, such as the previous word. Thus, if the word *Amede* is preceded by *Ato* then it is more likely to be a person name. Titles and designators have proved successful in NER and are called triggering words or triggers.

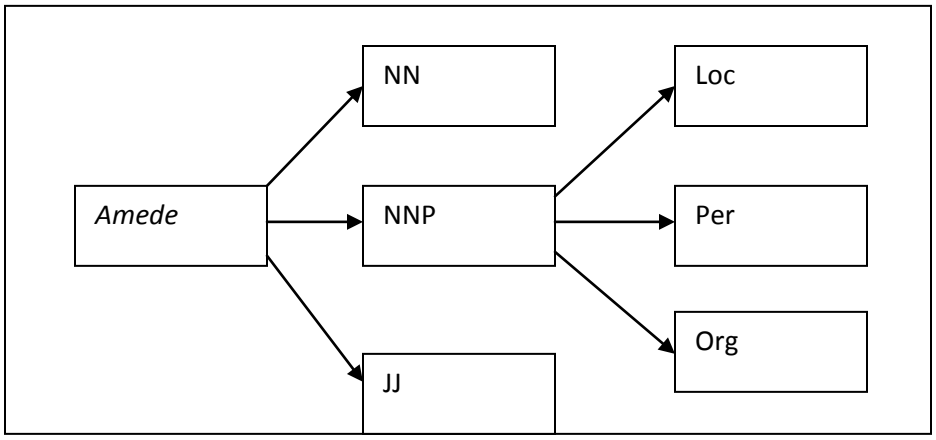

Figure 2.1 State- of- a sentence ambiguity example Shabib (2011) [1]

To be able to identify NEs successfully in such a situation, there would be a need to analyze the context, given that each word category would have a different context. For instance, an adjective typically cannot follow a verb without a determiner and also noun typically cannot be followed by an adjective in English. Thus, analyzing the surrounding lexical and POS context is crucial. If we have the means to generate the POS information correctly, then the problem would be solved to a large extent. However, it is not feasible as POS taggers rely on case information to tag proper nouns [3].

[1] proper noun (NNP), noun(NN), adjective (JJ), Location (L), person name(per), and organization (Org)

1.3 Statement of the problem

The motivation that initiated the researcher to conduct this research on NER is the advantageous and application areas of NERs. As it is stated in [4], NER is a key technology of Information Extraction and Open-Domain Question Answering.

According to [5] , Amharic 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 is used nationwide as the official language of the nation. Despite having large speaker population, the language has little computational linguistic resources. So, conducting a research on such kind of language is advisable to overcome the current shortage of computational linguistic resources.

Amharic is one of the under-resourced languages. Hence, there is no full-fledged Amharic Named Entity Recognition system that could contribute towards designing NLP. So that conducting Amharic NER with best performance will have a contribution to advance research in NLP for Amharic language.

There is only one attempt on Amharic named entity recognition. And further improvement regarding Amharic NER(ANER) has never been tried after the first ANER [6] attempt. Nationwide also there have been limited attempts for developing a NER of Ethiopian languages, Amharic by Moges[6] , Afan Oromo by Habtamu [7] and Mandefro [8] were attempted to develop Named Entity Recognitions. Moges [6] shown that some features played a great role in changing the performance of the Amharic NER system. So, in the proposed system the researcher tried to reconsider features that could increase the accuracy of Amharic NER.

According to [9] the best choice of feature combination chosen via the validation set correlated perfectly with the choice that has the best test set

performance, and in a system like [6] which followed Conditional Random Field (CRF) model, well-chosen features along with dictionary-based features tend to improve the CRF model's performance [10]**.** These shows considering other features would help to achieve a better performance. Therefore it is the present research to explore best feature sets for designing a high performance Amharic Named Entity Recognition.

Hence the research questions to be addressed have been the following

- What are the best feature sets for enhancing the performance of Amharic NER?
- Does a combination of different feature have an effect on the system?
- Which features are the optimal features sets in ANER?

1.4 Objectives of the study

General and specific objectives that have been tried to achieved are:

1.4.1 General objective

The general objective of this research is identifying the optimal feature sets that improve the performance of ANER system.

1.4.2 Specific objective

- Review literature so as to understand the role of named entity and approaches in NER
- Identify as many features as we can.
- Identify the optimal combination of the features that brings the best ANER Performance
- Selecting features that could be integrated with Amharic NER.

- Compare models that could help to improve the performance of Amharic NER.
- Evaluate performance of the proposed system by using a test dataset

1 .5 Significance of the study

NER was successfully incorporated into name searching systems to identify names in queries and the underlying data source for information retrieval [11]. Another important application is question answering, which was driven by one of Text Retrieval Conference's tracks. According to (Noguera et al. 2005) as cited by [3] NER is a core component of these systems and the effect of NER was measured in. The idea was to consider only documents that have at least one named entity provided in the query. It was found that NER could reduce the data returned by the IR system by 62% without any loss of information and by 92% with acceptable loss. Detecting NEs in a query was also success [3].

Conducting NER for Amharic is also very important for the development of Natural Language Processing (NLP)(such as, Speech synthesis, speech recognition, Question Answering, Information retrieval, Information extraction) [12]. In order to achieve such kind of advantages, developing a system with a high performance is unquestionable and to have a high performance ANER, finding the optimal features and identifying the best features combination are the main task of NERs.

Result of this research could be an input for different disciplines of NLP specifically in Amharic. This includes, Information extraction, Question answering, Ontology construction etc.

1.6 Scope of the study

Most common among marked categories are names of people, organizations and locations as well as temporal and numeric expression [13]. And within each category there could be additional sub categories. For example, for location may have city, mountain, river, sea, village etc. as its subcategory. In this study due

to time constraint only the three main NE classes names of people, organization and locations without their subcategories were taken.

1.7 Methodology

NER is a compound task that is to be done by different components at different steps. The research also has different developmental stages. First finding out the corpus from a news source was researchers' initial task. The learning methodology to be adopted requires customization, so it has its own computational task. Selecting the appropriate tools and techniques were the parallel task during all the stages.

1.7.1 Literature review

This work is conducted by reviewing a number of related literatures from books, journal articles, conference proceeding and the internet, so that choosing appropriate tools and methods were supervised. The large portions of reviewed materials are conference and journal articles.

1.7.2 Tools

A tokenizer during preprocessing of both training and testing data was developed by using Python programming languages. The reason python was chosen by the researchers was, it is easy for text processing. For NER part, Stanford NER tool is used.

As [12] describes machine learning techniques which can handle multiple features, a machine learning approach was analysed as an approach for NE learning process.

1.7.3 Data source and preparation

As it is mentioned above the tool was tested with freely available Amharic texts, which are collected from WALTA. Due to the availability of transliterated articles and acquiring those articles for freely, WALTA is chosen.

The collected articles were further edited manually by removing those sentences which does not have NEs at all. The resulting data is then tokenized and tagged manually in accordance with the Stanford standard. Finally, NE corpus of size 13538 was developed.

1.7.4 Test Procedure

The performance of methods and tools are measured based on the three main scoring approaches: precision, recall and F-measure. According to [13], these measures are the most commonly used for the assessment of statistical extraction systems and trace their origins back to the information retrieval discipline.

1.8 Organization of the work

The rest of this thesis is organized as follows. Chapter 2, literature review, concept related to named entity recognition is discussed. In the chapter background information of information extraction, feature sets types, machine learning approaches, CRF model and nature of Amharic language are described. In chapter 3 local and global literatures are reviewed and their achievements are discussed. chapter 4 contains design and implementation part of this work, experimentations that are conducted in the research are found in this chapter. Finally, conclusion and recommendations which are given based on the experimentation are found in chapter 5.

Chapter two

Literature review

2.1 Information Extraction (IE)

Currently, there is a considerable interest in using these technologies for information retrieval, since there is an increasing need to localize precise information in documents, for instance, as the answer to a question, rather than retrieving the entire document or a list of documents.

According to [14],IE systems extract domain-specific information from natural language text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies, and physical objects. Domain-specific extraction patterns (or something similar) are used to identify relevant information.

This early definition on information extraction is too limited and this definition is the same as current named entity extraction definition, Information extraction system should not necessarily be domain specific rather it should be domain independent or at least flexible to any domain with a minimum amount of engineering effort.

Some scholars are saying information extraction only extract a fragment information but other says it is the process of extracting fragment information from a corpus and then process it in a way that it gives meaning. According to [15],IE isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework. The goal of information extraction research is to build systems that find and link relevant information while ignoring extraneous and irrelevant information.

As [16] defines IE is the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making the information more suitable for information processing tasks.

2.2 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. For the first time the name NER system coined at MUC-6 but the system was applied in different names those are domain specific (extract and recognize company names). The term NER came after the increasing need of extracting specific information from a large corpus. To conduct a NER system defining the right tag for an entity is the most important task, there are conventions those defined tagging techniques of a named entity in a corpus. Message Understanding Conferences-6(MUC-6) and MUC-7 and Conference on Computational Natural Language Learning (CoNLL) are among the conventions that most researchers who are conducting on NER are currently using [17].

2.2.1 Message Understanding Conferences (MUC)

The Message Understanding Conferences (MUC) was a series of conferences organized by the (U.S.) National Institute of Standards and Technology and the U.S. Department of Defense Advanced Research Projects Agency. These events were held between 1987 and 1998 as part of the TIPSTER Text program [3] . It is believed that MUC seven conferences on IE contributed a lot for the development of NLP research particularly in IE. The MUC conferences focused on English, Chinese, Japanese and Spanish languages the three Identified MUC conference entities are ENMAX, TIMEX and NUMEX.

1. **ENMAX** contains, **Person name** - named person or family. Eg. Besufikad ,Alemu etc.

   Location name - name of politically or geographically defined location. Eg, Addis Ababa, Ethiopia etc.

   Organization name - named corporate governmental or other organizational entity. Eg. Addis Ababa University, International Leadership Institution etc.

2. **TIMEX** contains, **DATE** includes complete or partial date expression. ሃሙስ ጠዋት/Thursday Morning/ መጋቢት ፲፫ / March 22/ etc.

TIME includes complete or partial expression of time of day ቅዳሜ ማታ / Saturday night/

3. **NUMEX** contains- numeric expressions, monetary expressions and percentages; expressed in either numeric or alphabetic form.

   Money- monetary expression eg. 5 birr, $3 etc.

   Percent – percentage expression eg. 5%, 100% etc.

From the above listed MUC conference entity types researchers conducted by using either partially or complete availability of the types.

2.2.2 Conference on Computational Natural Language Learning

CoNLL is highly focused on the named entity recognition task. The named entity types classification somehow differs from MUC, the three sub-classes in MUC first class (NAMEX) are treated as an independent class and by introducing Miscellaneous names which includes product names, names of services, etc [18]. CoNLL categorized named entity types in four groups. In contrast to MUC the languages considered by CoNLL were Spanish and Dutch in 2002 and German and English in 2003.

2.3 Machine Learning

As (Grobelnik M and Mladeni´c D ,2005) cited on [19]different knowledge discovery methods have been adopted for the problem of semi-automated ontology construction including unsupervised, semi-supervised and supervised learning over a collection of text documents, using natural language processing to obtain semantic graph of a document, visualization of documents, information extraction to find relevant concepts, visualization of context of named entities in a document collection.

Machine learning algorithms can be classified as Supervised, Unsupervised and semi-supervised. The distinction comes from how the learner classifies data. In this section we will see the three kinds of machine learning.

2.3.1 Supervised learning (SL)

Supervised learning, the classes are predetermined. The classes are previously known. In a supervised learning, let the domain of instances be X and the domain of labels be Y. Let P(x, y) be an unknown joint probability distribution on instances and labels X × Y. Given a training sample $\{(x_i, y_i)\}$, supervised learning trains a function f : X → Y in some function family F, with the goal that f (x) predicts the true label y on future data x [20].

According to [21]the goal of supervised learning is to learn a mapping from x to y, given a training set made of pairs (x_i, y_i). Here, the $y_i \in Y$ are called the labels or targets of the examples x_i.

The process in supervised learning is that a certain part of data will be labeled with known classifications. The machine learner's task is to search for patterns and construct mathematical models. These models then are evaluated on the basis of their predictive capacity in relation to measures of variance in the data itself.

Supervised learning is the current dominant technique for addressing the NER. SL techniques include Hidden Markov Models (HMM), Decision Trees, Maximum

Entropy Models (ME), Support Vector Machines (SVM) and Conditional Random Fields (CRF). These are all variants of the SL approach, which typically feature a system that reads a large annotated corpus, memorizes lists of entities, and creates disambiguation rules based on discriminative features [22].

2.3.2 Unsupervised Learning (UL)

Unsupervised learning algorithms work on a training sample with n instances $\{\mathbf{x}_i\}_{i=1}^{n}$. There is no teacher providing supervision as to how individual instances should be handled, Common unsupervised learning tasks include, clustering, where the goal is to separate the n instances into groups, novelty detection, which identifies the few instances that are very different from the majority and dimensionality reduction, which aims to represent each instance with a lower dimensional feature vector while maintaining key characteristics of the training sample [20].

In unsupervised learning, there is no such supervisor and we only have input data. The aim is to find the regularities in the input. There is a structure to the input space such that certain patterns occur more often than others, and we want to see what generally happens and what does not [23].

According to [23] the goal of unsupervised learning is to group data into clusters . In fact, the basic task of unsupervised learning is to develop classification labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group. These groups are termed clusters, and there is a whole family of clustering machine learning techniques.

One can try to gather NEs from clustered groups based on context similarity. There are also other unsupervised methods. Basically, the techniques rely on lexical resources (e.g., WordNet), on lexical patterns, and on statistics computed on a large unannotated corpus [22].

In unsupervised, the machine is not told how the texts are grouped. Its task is to arrive at some grouping of the data. Unsupervised learners are not provided with classifications.

2.3.3 Semi- supervised Learning (SSL)

Semi-supervised machine learning came up with the combination of supervised and unsupervised learning. As [21]defines, SSL is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information but not necessarily for all examples. Often, this information will be the targets associated with some of the examples. In such kind of cases , the data set $X = (x_i)_{i \in [n]}$ can be divided into two parts: the points $X_l := (x_1, \ldots, x_l)$, for which labels $Y_l := (y_1, \ldots, y_l)$ are provided, and the points $X_u := (x_{l+1}, \ldots, x_{l+u})$, the labels of which are not known, and he called this a standard semi- supervised learning.

According to [20],SSl is not a half way rather it is an extension of SL and SSl also known as classification with labeled and unlabeled data (or partially labeled data), this is an extension to the supervised classification problem. The training data consists of both l labeled instances $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabeled instances $\{x_j\}_{j=l+1}^{l+u}$. One typically assumes that there is much more unlabeled data than labeled data, i.e., $u > l$. The goal of semi-supervised classification is to train a classifier f from both the labeled and unlabeled data, such that it is better than the supervised classifier trained on the labeled data alone [20].

2.4 Feature set 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. Authors supported the above idea with an 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 [7].

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”>”

But the above definition is working only for European language those are following capitalization to named entities (i.e. English), where as for other languages like Amharic, Hindu, Arabic …etc the above mentioned feature vector did not work.

One of the most challenging aspects in machine learning approaches to NLP problems is deciding on the optimal feature sets, there are different feature set and selecting the best features is not easy. In the baseline research (Moges 2010) the features have shown that a change and selection of them also was not easy. So, as [24] described when developing a NER for a specific language, having a large number of features at the beginning is advisable and it helps to come up with the optimal feature sets.

Different authors classified feature set in different ways, some grouped local knowledge and global knowledge [25],within each group there is also subgroups ;and others [24, 26] categorized the sub groups independently .

According to [25] local knowledge are feature which can be extracted from a token (word) being labeled and its surrounding context.

Token (word) – based features contains suffixes, prefixes and orthographic features (shape) of the current token. Context – based features are Previous and next words of a particular word. The performance we get from token based and context based features are different. They are also domain specific, as [25]describes context based feature outperform than word (token) in biomedical entities and in newswire the result is vice versa.

External knowledge features are additional features that determine the entity of a token beyond the context and word based feature. According to [25] External knowledge features are part-of-speech (POS) tags, shallow parsing and gazetteers.

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 [7]. Pos tagging by itself is challenging and time consuming due to this the baseline and other NERs are used manually tagged corpus. Because of the aforementioned reasons [27]developed without it.

Phrasal clustering- is clustering a corpus in to n clusters by using different clustering techniques and use the phrase to identify tokens entity.

Gazetteers- list of predefined names of persons, locations, organizations etc. Even though developing gazetteers is not easy working on the available list is also difficult. Different problems are raised from the misuse of gazetteers. According to [28] semantic ambiguities are occurred when we use gazetteers, there is a situation that a single name could be person name , organization name or location name. For example, *Alem* can be name of a person, organization or location. In such kind of scenario it is difficult to decide the right entity for a token.

The other challenge of using a well prepared gazetteer is that they are memory intensive that comes from people names and organization names are different due to this there will be numerous names in the list but the occurrence of those tokens in a corpus is rare sometimes they might not be presented once in the whole document and having those unused words in the corpus is wastage of our memory space. In addition to this a large coverage gazetteer would not guarantee a better accuracy. As [3]said this is due to the fact that person names are mostly found in form of nouns and adjectives. Whereas rearranging the list in the gazetteers could be improve the performance, according to [13] rearranging means selecting the most important out of the predefined list and adding the remained names that believed to bring a better performance.

2.5 Conditional Random Fields (CRF)

Conditional Random Fields can be understood as the sequence version of Maximum Entropy Models, that means they are also discriminative models [29]. The Conditional Random Fields (CRF) are probabilistic models for computing the probability p (Y|X) of a possible output Y = (y_1,…y_n) ε Y^n given the input X = (x_1 …x_n) ε X^n which is also called the observation. In this section CRF which is used in our ANER model is discussed in detail.

According to [30] linear chain CRFs is a special form of a CRF, which is structured as a linear chain that models the output variables as a sequence.

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

Where, j specifies the position in the input sequence x, but the weights λ_i are not dependent on the position j. This technique, known as parameter tying, is applied to ensure a specified set of variables to have the same value. n specifies the length of the sentence, m specifies the number of feature templates.

The normalization to [0; 1] is given by

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y}\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)\right) . \quad 2.$$

As it is indicated in the above two equations, features are dependent on the label sequence and herewith on the state transitions in the finite state automaton. So it is important to point out that only a subset of all features f_i is used in every transition in the model.

According to [30],the strategy to build a linear-chain CRF summarized in the following way:

1. Construct Stochastic finite state automaton (SFSA) S= (S, T) out of the set of states S (with transition T=$(s, \dot{s}) \in S^2$). It can be fully connected but it is also possible to forbid some transitions because the decision made in

this stage is depending on the training data and the transitions contained there.

2. Specify a set of features templates $F = \{g_1(\vec{x}, j), \ldots, g_h(\vec{x}, j)\}$ on the input sequence. This helps for the generating features f_i.

   $$g(x,j) = \begin{cases} 1 & if\ x_j = names \\ 0 & else \end{cases}$$

3. Generate set of features $\mathcal{F} = \{\forall s, \dot{s} \in S.\ \forall g_o \in F : f_k(s, \dot{s}, g_o)\}$

There are two problems in a linear chain CRFs that plays a great role in varying the performance of NERs.

The first one is, there is a given observation x and a CRF M: here the problem is finding the most probable label sequence Y. This problem is the most common application of a conditional random field to find a label sequence for an observation.

The other one is, given label sequences Y and observation sequences X: How to find parameters of a CRF M to maximize P (Y|X, M) are also a problem. This problem is a question of how to train to adjust the parameters of M which are especially the feature weights. In our system we have tried to address both of the above mentioned problems of linear chain CRF model.

2.5.1 CRF training

To train a CRF we need labeled data sequences. According to [31],CRFs training on unlabeled data is difficult. The labeled sequences are, { $(x^{(1)}, y^{(1)}),\ldots,(x^{(n)}, y^{(n)})$}, where $x^{(1)}=x_{1:N1}$ the first observation sequence, and so on. The main objective for parameter learning is to maximize the conditional likelihood of the training data.

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

$$= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)\right)}{\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right)} \right) \right] \quad 3.$$

During training there are features that are more than enough to determine the tag of a specific term. That means over fitting could occur and to control such kind of scenario over fitting likelihood is penalized with the term $-\sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma^2}$. The parameter δ^2 models the tradeoff between fitting exactly the observed feature frequencies and the squared norm of the weight vector. The smaller the values are, the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced. The notation of the likelihood function L(T) is reorganized as:

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

$$= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right) \right) - \right.$$

$$\left. - \log \left(\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right) \right) \right] - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

$$= \underbrace{\sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)}_{\mathcal{A}} -$$

$$\underbrace{- \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \log \underbrace{\left(\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right) \right)}_{Z_{\vec{\lambda}}(\vec{x})} - \underbrace{\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}}_{\mathcal{C}}}_{\mathcal{B}} . \quad 4.$$

The partial derivations of L(T) by the weights for parts A,B and C are computed separately.

Derivation for part A:

$$\frac{\partial}{\partial \lambda_k} \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{j=1}^{n} f_k\Big(y_{j-1}, y_j, \vec{x}, j\Big) \tag{5.}$$

Derivation for part B, which corresponds to the normalization, is given by:

$$\begin{aligned}
\frac{\partial}{\partial \lambda_k} \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \log Z_{\vec{\lambda}}(\vec{x}) &= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \frac{\partial Z_{\vec{\lambda}}(\vec{x})}{\partial \lambda_k} \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \sum_{\vec{y}\,'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\Big(y'_{j-1}, y'_j, \vec{x}, j\Big)\right) \cdot \\
&\qquad \cdot \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j) \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}\,'\in\mathcal{Y}} \underbrace{\frac{1}{Z_{\vec{\lambda}}(\vec{x})} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\Big(y'_{j-1}, y'_j, \vec{x}, j\Big)\right)}_{=p_{\vec{\lambda}}(\vec{y}|\vec{x})} \cdot \\
&\qquad \cdot \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j) \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}\in\mathcal{Y}} p_{\vec{\lambda}}(\vec{y}\,'|\vec{x}) \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j)
\end{aligned} \tag{6.}$$

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

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \tag{7.}$$

Equation 5, the derivation of part A, is the expected value under the empirical distribution of a feature f_i:

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \tag{8.}$$

Accordingly, equation 6, the derivation of part B, is the expectation under the model distribution:

$$E(f_i) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}'\in\mathcal{Y}} p_{\vec{\lambda}}(\vec{y}'|\vec{x}) \sum_{j=1}^{n} f_i(y'_{j-1}, y'_j, \vec{x}, j) \quad 9.$$

The partial derivations of L(T) can also be interpreted as

$$\frac{\partial \mathcal{L}(\mathcal{T})}{\partial \lambda_k} = \tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} \quad 10.$$

And the maximum likelihood method finds the parameter by equating the first derivation to 0 as:

$$\tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} = 0 \quad 11.$$

Computing $\tilde{E}(f_i)$ is easily done by counting how often each feature occurs in the training data. Computing $E(f_i)$ directly is impractical because of the high number of possible label sequences. 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 complex to explain in this study [31].

2.6 Named entity recognition tools

There are tools which helps to label NEs, Natural language tool kit (NLTK), Stanford named entity recognition, GATE, Lingpipe… etc. NLTK is a python platform that is more suitable for Part of Speech (POS) tagging. Stanford NER is a Java implementation of a Named Entity Recognizer developed by Stanford University. When Stanford started NER their scope was focused on identifying Biological terms (genes and proteins) from biomedical papers [32]. After their first attempt, they have expanded their scope by increasing the classes and also with regard to features the model they have used changed from time to time till

they implemented a sequence classifier to the unseen domains (i.e. CRF). We have used this customizable and open source NER tool for labeling our data.

Lingpipe is also 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 [33]. [6, 8] are used this tool.

2.7 Amharic language structure

In this section we have discussed language structure of Amharic with regard to our focused area, which is NE in Amharic. The section describes the background of Amharic script, word classes and characteristics of Amharic NEs. Even though, there are different word classes in Amharic we have only focused on Nouns and pronouns which are related to our work.

According to [5], 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

The present Amharic writing system was adopted from the Ge'ez writing system. Ge'ez, which belongs to the class of Semitic languages, was the language of literature in Ethiopia in earlier times [34]. The ancient Sabaean script is in turn attributed to the source of the Ge'ez script. However, as [34] explains, the number of symbols in the original Sabaean script and their shapes have changed into Ge'ez and then later on into Amharic. Moreover, some new

symbols have been added to Amharic. The current Amharic writing are combination of all Ge`ez fidel and some added characters.

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. It consists of 33 consonants, giving 7*33=231 syllable patterns, or fidels. In addition to the 231 characters, there are other non-standard alphabets which contain special features usually representing labialization. Each alphabet represents a consonant together with its vowel. The vowels are used to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order. Refer to appendix-1 for a complete list of the symbols [35].

Word classes of Amharic language are classified in to different categories by different scholars. As Mersehazen(1935 e.c), cited on [36] put the word classes as eight. Those are preposition, noun, conjunction, interjection, verb, adjective, pronoun and verb. Whereas, the recent works (Baye, 1987 and 97) as cited by [36], reduced the previous classes in to five word classes. The reduction of Amharic word classes are based on different reasons, Baye's reduction of Mersehazen's classification seems based on the role of words in syntax, which means by considering the clear role of words in Amharic grammar.

2.7.1 The Noun Class

Like English, Amharic nouns are words used to name or identify any of a class of things, people, places, organization or ideas or a particular one of these [37]. Formations of nouns are either simple, derivative or compounds.

As [34] Discusses, the formation of nouns are described as follows:

Simple forms; consisting of two, three, or four letters.

A. Bilateral.

a. Ending in the second order:

ሙሉ፡ full.

ጥሩ፡ good

ክፉ፡ bad

These all words are originating from or representing the passive participle.

b. Ending in the third order, generally signifying an agent:

መሪ ፡ guide

ሰፊ : wide

ሰሪ : workman

c. Ending in the fourth order:

ውኃ፡ water

ስጋ: flesh

ካራ: knife

d. Ending in the fifth order:

ቅቤ፡ butter, oil.

ጥሬ: genuine, original.

በሬ : ox.

e. Ending in the sixth order.

ቀን ፡ day

ላም : cow

ሙዝ፡ banana.

Most of bilateral nouns are ending in these order and they are numerous.

f. Ending in the seventh order:

ጉዞ ፡ a day's march.

ጎጆ ፡ small thatched house.

ቆሎ ፡ fried grain.

In triliteral and bilateral, [34] listed out different nouns. Noun words with more than two alphabets (fidel*)* are discussed in detail.[2]

Compound nouns: are nouns which are written as two separate words and sometimes as a single word [38]. E.g. " ቤተ-መንግስት" which mean palace. "ትምህርት ቤት" school "ፅህፈት ቤት" and "ፅፈትቤት" which both mean secretariat office.

Derivative nouns- are nouns which repeat themselves in order to pluralize the quantity they are expressing for. E.g.

ፍሬ	ፍሬ-ኣ- ፍሬ	ፍራፍሬ
ቁስ	ቁስ-ኣ-ቁስ	ቁሳቁስ
ትል	ትል-ኣ-ትል	ትላትል

2.7.2 Pronouns

According to [34] pronouns are divided as in other languages, into:

1. Personal
2. Possessive
3. Demonstrative
4. Relative
5. Interrogative
6. Reflective and

7. Distributive Pronouns.

[2] C. W. Isenberg, *Grammar of The Amharic Language*. London: Richard Watts 1976.

These classes are grouped into two categories, namely, Separable and Inseparable Pronouns.

Personal
Possessive
Demonstrative
Interrogative
} Separable

Relative
Distributive
} Inseparable

2.7.2.1 Separable Personal Pronouns

They are three for the Singular, and three for the plural. The singular has some peculiarities. The first Person has not the gender expressed, whereas, the second and third have distinct forms for the Masculine and for the Feminine gender. The second person has, besides, three distinctions of honor.

	Singular			Plural
	Masculine	common	Feminine	common
1 Person		እኔ: I		እኛ : we.
2 Person	አንተ: you		አንች: / አንቺ: } thou	እናንተ: you.
		አንቱ / እርስዎ } you		
3 Person.	እርሱ: he,it.		እርስዋ: she, it.	እርሳቸው: they.

The above structure is Amharic separable personal pronoun formation and Detailed discussion on Amharic pronouns are found in [34].

2.7.3 Amharic Named Entities

According to [39], NEs are phrases that contain the names of persons, organizations and locations. In the expression named entity, the word named restricts the task to those entities for which one or many rigid designators stands for the referent.

Person names (PERSON) - name of a person including his/her father and grandfather

Location name (LOCATION) – name of geographical entities like cities, regions, country.

Organization name (ORGANIZATION) – name of entities like companies, institutions and organizations.

Amharic has no case information like English and other Latin character languages which ease identification of NEs. It is one of those properties used in the tool (Stanford Named Entity Recognition), but due to the previously mentioned property of Amharic we have excluded capitalization feature.

There are words that mostly co-occurred with NEs, Clue words are useful properties in identifying NE and that mostly precede NE words and sometimes they succeeded NEs. Clue words for each NE classes are different, some of them for PERSON are:

ፕሬዚዳንት	President
አቶ	Mr.
ወ/ሮ	Mrs.

ፕሬዚዳንት ግርማ ወ/ጊዮርጊስ	President Girma W/Giorgis
አቶ በዓሉ ግርማ	Mr. Bealu Girma
ወ/ሮ አዜብ	Mrs. Azeb

Organization NEs are most of the time succeeded with the following clue words.

ካፒታል ሆቴል	Capital Hotel
ሐረማያ ዩኒቨርሲቲ	Haramaya University
ፕሬዚዳንት ቢሮ	Office of president

የተባበሩት መንግስታት ድርጅት United Nations Organization

Location NE classes have the same properties with Organization. In addition to clue words, Amharic NEs present with affixes.

Affixes are classified in to three, Prefix, Suffix and Infix.

Prefixes, suffixes and Infixes are sets of letters that are added to the beginning or end of another word. They are not words in their own right and cannot stand on their own in a sentence.

E.g. □□□□ □□□□□ □□□□□ □□□□□ □□□□

Esayas Afewerki is president of Eriteria.

□□□□□ □□□□□ □□□□□....

Esayas Afewrki's government…

On the first sentence /□□□□ □□□□□ / is a PERSON name, whereas, when we add the prefix /□/ the name changes it class to ORGANIZATION.

E.g. አዲስ አበባ ዩኒቨርሲቲ በኢትዮጵያ ዋና ከተማ ውስጥ ይገኛል፡፡

Addis Ababa University is found at the capital city of Ethiopia.

ከአዲስ አበባ ዩኒቨርሲቲ እስከ…

From Addis Ababa University….

Here in the first sentence the word /*አዲስ አበባ ዩኒቨርሲቲ* / is an ORGANIZATION name, but in the second form of the word /*ከአዲስ አበባ ዩኒቨርሲቲ* / becomes LOCATION name. The change comes after the addition of prefix /ከ/. N.B the second case is not always true, because the prefix by itself is not sufficient to identify NE classes. For eg.

አዲስ አበባ ዩኒቨርሲቲ በኢትዮጵያ ዋና ከተማ ውስጥ ይገኛል፡፡

Addis Ababa University is found at the capital city of Ethiopia.

ከአዲስ አበባ ዩኒቨርሲቲ የተማሪዎች መማክርት…

From the student council of Addis Ababa University…

In such cases the name/አዲስ አበባ ዩኒቨርሲቲ/ with prefix /ከ/ and the name /አዲስ አበባ ዩኒቨርሲቲ /in the first sentence have the same NE class, ORGANIZATION.

2.7.4 NER challenges for Amharic

Several studies indicated that NER is a difficult task and a number of challenges are still not addressed because of many reasons: Ambiguity, Agglutinative Nature of the Language, Spelling Variations and A Nested Named Entities**.**

2.7.4.1 Ambiguity

In Amharic there are proper names that have more than one NE class. ፀሐይ (*Sun)* can be person's name and it is used to name Sun. Amede, H/Giorgis, Haile Gebresilase ,Abune Petros and Belay Zeleke are some of the names that are used for both location and person's name in Amharic. During model creation, handling ambiguity words are performed with a feature that can learn their contextual pattern in the sentence.

2.7.4.2 Agglutinative Nature of the Language

Agglutinative property means that some additional features can be attached to the word to add more complex meaning. According to [7],most of the time NE root words cannot be found in a document, rather they are found in their morphological derivations that are formed by affixes. As we have discussed in section 2.7.3 Amharic Named Entities prefixes helps to identify class of a word in Amharic. For example, when the prefix, ከ- adds to ኢትዮጵያ (Ethiopia) it becomes ከኢትዮጵያ (From Ethiopia) and we can used such kind of affix as a feature to classify that word under LOCATION class.

2.7.4.3 Spelling Variations

In Amharic there are words which can be written in different formats. For instance, "ፀሐይ" and "ጸሐይ", "ፀሀይ" and "ጸሀይ" all mean the same, "sun" although they are written differently (indicating alternate use of ፀ and ጸ, and ሀ and ሐ. In our research we have tried to handle such kind of language challenges by using transliterated data, in transliterated corpus all the afore mention names are written in a single format, *tShay* (Sun) and controlling spelling variations in

NERs ease extracting features and applying it in all words that have found in the same format.

2.8 A nested NEs

It is a word that are formed by a combination of two names, but identifying NE class of a word without contextual consideration leads to wrong classification of NE class and less NER system performance. A word can be classified any of the three NE feature, but when it comes with some words it changes the previous NE class, such kind of problems are common in Amharic. For instance, አዲስ አበባ ዩኒቨርሲቲ Addis Ababa is location name, but when the word ዩኒቨርሲቲ (University) ከተማ አስተዳድር (City Government) and some others are added it becomes Organization names. አዲስ አበባ ዩኒቨርሲቲ (Addis Ababa University) and አዲስ አበባ ከተማ አስተዳድር (City Administration of Addis Ababa). Whereas studying the co-occurred names during model creation overcome challenges that existed in nested NEs.

Chapter three

Related works

There have been different NER systems development attempts taken by different authors throughout the world. In this section an overview of NER will be discussed, by categorizing works on domestic and foreign language. In the domestic languages, one Amharic [6] and two Afan Oromo [8] [7] named entity recognition systems have been developed.

[6] Used CRF to develop a NER system in Amharic with internal and external features, Context features, Part Of Speech tags of tokens, prefix and suffix. The corpus contains 10405 tokens, was split into training, developing and testing data set. To conduct the experiments the researcher used tagged corpuses that are found from news articles, they followed supervised learning method. During the experiment identification of the most relevant features for Amharic NER were attempted and the reported accuracy at first was F-1 73.47% with all of features. When the system was tested by removing POS tag, Prefix and Suffix the performance was 69.70%, 74.61% and70.65%, respectively. The reason,[6] concluded, was that POS tag and Suffix are most relevant to identify Amharic NEs. Even though their research is conducted on an annotated corpus that are taken from WALTA news agency, POS tagging was trained by using Ling pipe tool it is a customizable environment developed for English.

[8] Used a supervised machine learning method and CRF model to develop a NER system in the Oromo language. The features used are both internal and external, normalized tokens, part-of-speech tags, word-shape features, position features, and token prefixes and suffixes. Adopted from CoNLL(Conference on Computational Natural Language Learning), four types of Oromo language NEs are identified: Person, Location, Organization and Miscellaneous. As they

described due to compatibility issue with previously proposed work for Oromo language POS tagging which is external feature in their work they have used ling pipe tool that is originally developed to English. The processing was classified in to pre-processing, training and recognition phase. They have reported that to all processing they have used news based Afan oromo corpus which consists of 23,000 tokens from "*Bariisaa*" and "*Kallacha Oromiyaa"* news papers. After generating the model they reported that on the first experiment the performance obtained are 77.41% Recall, 75.80% Precision and 76.60% F1-measure. They have conducted their experiments by adding the corpus size and reducing features. They found that recall and F1- measure increase while precision reduced while removing feature in the experiment has shown an inverse change in all performance measurements. Finally [8]reported that features play a vital role than the change in the training data size.

The other attempt in the Oromo language is the development of Afaan Oromo Named Entity Recognition (AONER) system. As they reported, AONER used the same method and model that have been used by [8]. The corpus in [7] is also news articles that is collected from Oromia Radio and Television organization, it contains 30,402 tokens. Features used to identify the tokens in to Organization, Person and Location formats are current word, N-grams, previous word, next word, maximum n-gram length, current word shape and surrounding word shape sequence features. The data is split in to training and test set to conduct the experiment. [7] conducted three experiments those are based on current word, the use of n-gram(left/right word) and word shape sequence to identify and detect the boundaries of named entities. As they reported, the overall performance achieved by using the above listed features is F1- measure of 84.71 on test data set. The identified features that have the greatest significance during the experiment are categorized regarding their indication of named entity type. For Named entity type they found that, contextual features (previous, next word, two word previous, and two words next to current word) are a good indicators, clue words are good in identifying all named entity types due to the

reason that has the best predictive value for classifying the NEs and finally as [8]concludes they have also summarized their work, combinations of context and word shape sequence are very important for Afaan Oromo NER.

[3]Used first-order Markov HM-SVM with linear kernel and Expectation Maximization(EM) on the Arabic NER with different feature sets, Tag of other occurrences (GLB), Previous Class (C-1), Trigger(TRG),list of unambiguous names(UNQ),list of most frequent non-person bigrams(BIAMB),effective preceding unigrams(UNIG), Most frequent non-person tokens(AMB), POS tagging (POS), Gazetteers (GAZ) and Lexical (LEX). The data used was ACE corpus is in the form of raw text files of news stories with stand-off annotation in XML files , from the corpus they annotated 70,000 words of the Arabic Tree Bank with their named entity classes. As they reported, ten experiments have been taken place by interchanging the above listed features. they have tested the corpus with two algorithms they have selected, due to the use of different features in the two learning algorithms they did not want to compare the two algorithms but the result found to HM-SVM and EM are 74.49and 69.18, respectively. As [3]reported, best average performance was on the location class, as it was easier in terms of the context of POS tags and also due to the repetition of location entities in the corpus and worst result was on the organization class, since there was no gazetteer employed and also due to the syntactic structure of their naming systems that make them look like any other Arabic phrase. After repeated experiment the identified features those have a positive impact to all NEs are the class of other occurrences of the word being processed together with a list of triggers, class of the previous word and POS

[40] Used a hybrid (Gazetteers and Hidden Markove) model on Indian Languages. They identified four named entities; Person (PER), location (LOC), temple, River and rest are assigning other tag. During the experiment they have used limited sentences that are domain specific (Tourism) corpus. First their experiment tested the corpus by each model independently and scored 40.13%

and 97.3% for Gazetteers and Hidden Markove model, respectively. Hence [19] described that the performance is varied but the corpus size they have used for the two experiments is different, 100 (Gazzetteers) and 40 (Markove) sentences. Finally they concluded that when they applied the hybrid model it scored 98.375%, with this findings [19] reported, that this NER system with high accuracy is built, then this will give way to NER in all the Indian Languages and further an efficient language independent based approach can be used to perform NER on a single system for all the Indian Languages.

[2] Used supervised method with both annotated and unannotated data that was manually tagged at the University of Antwerp. sixteen systems are participated in CoNLL-2003 shared task , Maximum Entropy model is the most used model by a participants and Conditional Random Fields also used by one of the participants , the participants were used different models separately and with combination one with the other. Four types of named entities are tried to detect, persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. The CoNLL-2003 named entity data consists of eight files covering two languages: English and German. For each of the languages there is a training file, a development file, a test file and a large file with unannotated data The English data was taken from the Reuters Corpus and The portion of data that was used for Germen data, was extracted from the German newspaper Frankfurter Rundshau. Main features used by the sixteen systems that participated in the CoNLL-2003 shared task sorted by performance on the English test data. affix information (n-grams, bag of words, global case information, chunk tags, global document information, gazetteers, lexical features, orthographic information, orthographic patterns (like Aa0), part-of-speech tags, previously predicted NE tags, flag signing that the word is between quotes and trigger words. Finally they concluded that, the best performance for both languages has been obtained by a combined learning system that used Maximum Entropy Models, transformation-based learning, Hidden Markov Models as well as robust risk minimization

Conclusion

Literatures that we have reviewed have shown that all of them have followed supervised learning method and statistical approaches, but they have used different features. It clearly shown in [8] and [7] that both of them used the same model and similar corpus with different features from which they have got different performance. From [2] we realized that applications of machine learning models interchangeably for the same or different language would not bring a significant change. Whereas an effective selection, identification and understanding of features and nature of languages are the most important points to come up with a high performance NER system.

CHAPTER FOUR

Design and implementation of the system

4.1 Design of ANER

In this section we have discussed the overall design of our system, Amharic Named Entity Recognizer (ANER). The first part of this section discusses about our data sets which is used for training and testing phases. The second section contains approach we have followed in the study. Then the general overview of the proposed system architecture from the perspective of the system's flow of operations in the form of processes is discussed. Finally, detailed explanation of phases along with subcomponents included in each phase and our system performance measurement are presented.

4.1.1 Data sets

As it has been defined in [6],[7]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. In this ANER, we have used transliterated 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. But we have chosen to use the transliterated one, as we have discussed in section 2.7.4.3, using transliterated corpus reduced spelling variations which are common in Amharic script (fidel) writing. 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. The corpus 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”

and this makes direct use of the corpus difficult. So, removing the string has taken place before going to the main preprocessing phase.

4.1.2 Corpus preparation for ANER

In our system we need two types of data, training and testing data. Training data are used to train the machine learning component. Then after, conducting training and creating a Model, the system has tested on testing data to evaluate the performance of the system. Before going to prepare our data we have been tried to get previously annotated data by [6]; both the researchers and the respective department were not able to provide us the needed data, after several attempt finally we gone to prepare our own data.

By using simple random sampling technique, 13538 tokens from WIC corpus were selected for both phases. During sample selection sentences with a minimum of one NE have been selected, this is with the assumption of if there is no NE in a training data, finding the pattern and generating a good model will be tough and testing such kind of model may have degrade the performance of a system.

After selecting the corpus and conducting preprocessing tasks, removing manually tagged POS and tagging with named entities based on Stanford notation were conducted. According to the format every word in a sentence will be kept on one line together with the NE tag separated by a tab space.

4.1.2.1 Data format

yemaIkelu O
astebabari O
ato O
belayneh PERSON
ayalEw PERSON
lewalta ORGANIZATION
InformExn ORGANIZATION
maIkel ORGANIZATION
IndegeleSut O
maIkelu O
" O
abobako O
" O
yemil O
syamE O
yeteseTewn O
yebeqolo O
zerna O
" O
1xi O
107 O
" O
yeteseNewn O
yemaxla O
zer O
yageNew O
balefew O
and O
amet O
bakahEdew O
mrmr O
new O
:: O

The data contains entity of three types: persons (PERSON), organizations (ORGANIZATION), and locations (LOCATION). Words tagged with O (Others) are words which do not belong to the above three named entities.

4.1.3 Approach Used

From the related works that we have reviewed [22], supervised learning is the current dominant technique for addressing the NER and ANER is designed by following this approach. The machine learning component learns from annotated training data. Conditional Random Fields (CRF) algorithm is the machine learning component we have used. A CRF algorithm has the efficiency of dealing with non-independent, diverse and overlapping features. It has also the capability of overcoming the label bias problem. These two factors make CRF algorithm better suited for NER. The preprocessed annotated data is given to the training phase to create a model, then, testing phase, which is the recognition of the named entities using the model that is generated by the training phase are performed. Figure 4.1 describes our architecture.

4.1.4 Architecture of ANER

Under this section, the architecture of the system will be presented. Figure 4.1 shows the overall functionality of the Amharic Named Entity Recognition system. There are three main phases in our ANER system, Preprocessing, training, and testing. These three major phases might be used in different NER and other machine learning systems, but tasks which are taken place within each phase are different and this is caused by the data we used and the procedure that a researcher has chosen. Because of the above reasons, there is no single architecture that we all should follow. With this assumption our ANER architecture is described as follows.

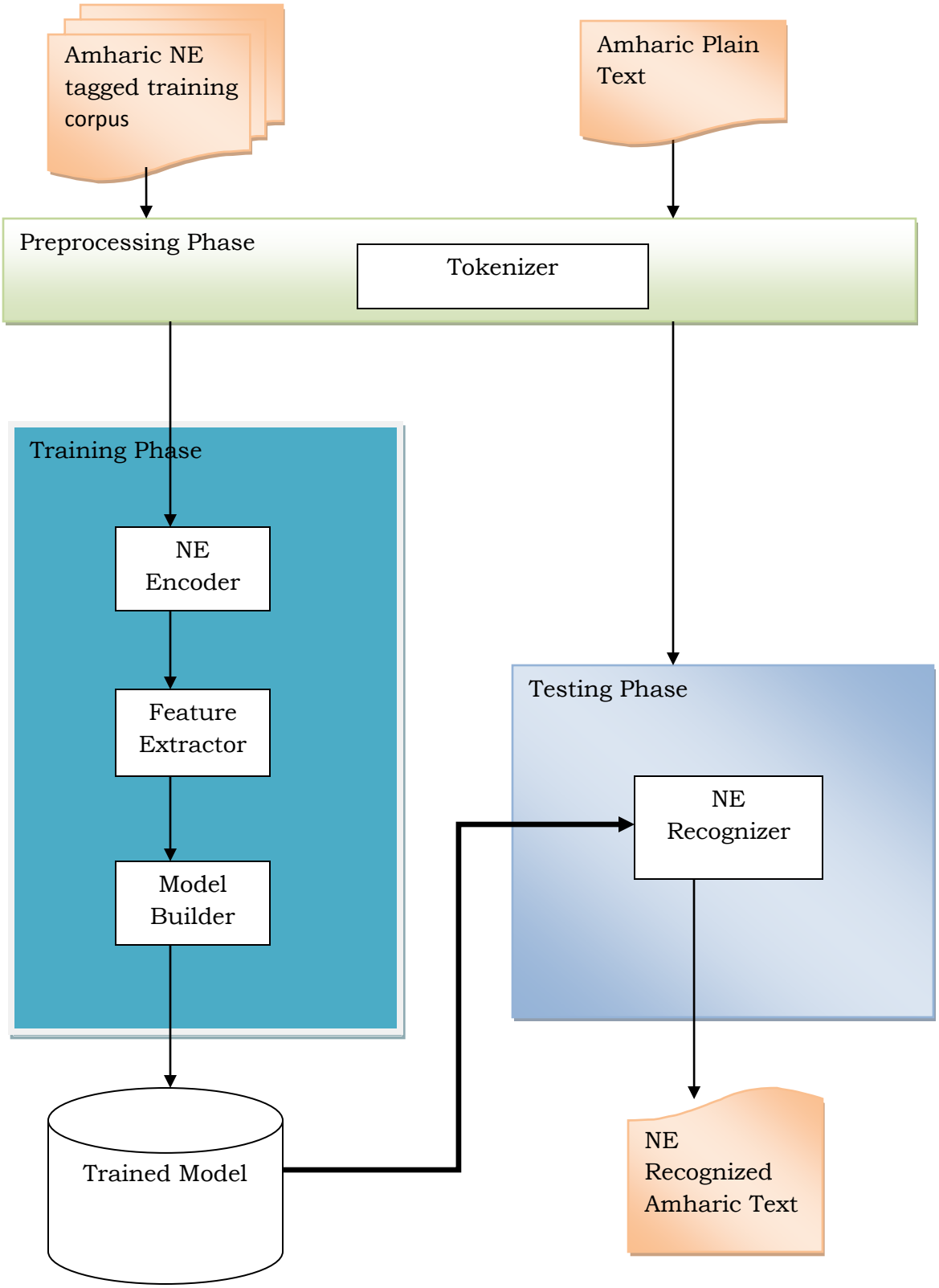

Figure 4.1 Architecture of ANER.

4.1.4.1 Processes of ANER

ANER is designed in a manner that, first it learns properties and parameters associated with NEs from the training data. It then receives input plain text and predicts the possible NEs out of the plain text. The architecture has two processes: learning process and prediction process.

4.1.4.2 Learning process

Training is handled by components in the learning process. Preprocessing is initially performed on the training corpus. The corpus is tagged based on Stanford NER format. Then it is passed to the NE encoder which identifies a token and its corresponding NE tag through encoding. The token and tag sequence generated by the NE encoder is handled to the Feature Extractor. Feature Extractor extracts necessary features to identify NEs based on the generated token and tag sequence, the extracted features will then be used as an input to the Model Builder. After the above all fulfilled, the builder begins the model building procedure to generate a trained model. Generally, here in the learning process, based on the training corpus we have generated a model that will be used to predict NE classes of testing.

Pseudo code for learning process:

For *all tokens in the training data*

Activate *previous word sequence feature*

Based on the specified maximum length of left and right window size

fExtractor .extract(previous word feature for the current token and set as 1)

Activate *next word sequence feature*

Based on the specified maximum length of left and right window size

fExtractor .extract(next word feature for the current token and set as 1)

Activate *word pair feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(word pair feature for the current token and set as 1).

***Activate** word shape feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(word shape feature for the token and set as 1).

***Activate** NE tag feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(NE tag feature for the current token and set as 1).

fExtractor.append(the NE tag of the previous token and set as 1).

fExtractor.extract(the NE tag of the next token and set as 1).

***Activate** prefix and Suffix feature*

Based on the specified maximum length of prefixes and suffixes

fExtractor.extract (prefix & suffix feature for the current token and set as 1).

End For

Return fExtractor().

4.1.4.3 Prediction process

Here the plain text is preprocessed. Tokenization is the only preprocessing task performed on the input plain text. The tokenized data is then passed to the NE Recognizer. The trained model will assist the NE Recognizer to perform such tasks as calling stored features and NE tagging on the input data. The NE Recognizer recognizes NEs out of input text with the aid of the trained model. The result, NE recognized Amharic text is then supplied as an output.

4.1.4.4 Preprocessing

4.1.4.4.1 Tokenization

Tokenization can basically be defined as the process of breaking up a text into its part of elements, tokens. In Amharic, tokenization process is easier since every word is separated by a space. Tokenization is done on the input plain text during the prediction process. Tokenization on the training corpus is also done during corpus development. The tokenization takes the input text supplied from a user and tokenizes it into a sequence of tokens that can make it easy to recognize NEs.

4.1.4.5 Training phase

Training phase contains necessary components that are used in the process of training the machine learning component. The components within this phase operate on the data from the training corpus and they are the NE Encoder, Feature Extractor and the Model Builder.

4.1.4.5.1 NE encoder

This is a process that provides information to feature extractor by identifying a token and its tag. From the training data the encoder analyzes which one is a token and which one is a tag then supplies to the next process. Hence, the encoder provides important clue to feature extractor, setting it wrongly also leads to wrong model creation. To avoid this we have kept the consistency between corpus and NE encoder.

4.1.4.5.2 Feature extractor

As we have discussed in chapter two, Features are properties of a text that are used to provide necessary information associated to a given NE and increase the confidence level of predicting a token as a NE. ANER's feature extractor is responsible for identifying and extracting all the necessary features from the training data. It is one of the essential components that supply necessary information (features) during the building of the model. The feature extractor is designed to extract features from the training data and store them for future use.

The feature extractor that we have used, CRF algorithm, performs repeated feature extraction from the same context for different positions in the input. All the extracted features are supplied to the model builder which will predict the parameters of the model. To identify Amharic NEs we have used: Word Prefix and suffix, word shape features, word position features with a different window sizes and the detailed feature combination used are described in section 4.2. As we have discussed in chapter two POS tag feature is used in different NER systems and they have scored various performances, but with regard to the tool we have chosen (Stanford), the features used in our work are very similar to POS tags. Due to this reason we have focused on using other features that predict NEs of Amharic.

4.1.4.5.3 Model Builder

The main concern with the machine learning component is to build a trained model that will be used in prediction. Building a model is the component designed to estimate the model Coefficients and then build a trained model. Estimation is taken place based on the input from the training corpus that is supplied by extracted features which is generated from feature extractor. The model builder is designed to perform all the calculations we have discussed in section 2.5 and generated a trained model that will be used in prediction phase.

4.1.4.6 Prediction phase

This prediction or recognition is the final phase in NER system. It is a process of recognizing NEs from the given preprocessed token. There are two sub tasks are taken place in this phase, detection and classification.

4.1.4.6.1 Detection

Detection is the first task which finds candidate tokens that can be NEs from the tokenized plain text. By using the knowledge acquired from the model, detection is performed. The knowledge in model builder contain features that are extracted and stored during the training phase, are supplied to the recognizer to identify NEs from the given plain text. Identification of NEs are performed based on the calculated probability.

4.1.4.6 .2 Classification

After detection of NEs for the candidate tokens are performed, classification process is preceded by tagging the detected candidate tokens. During classification each token in the given plain text will be tagged with their possible NE tags using Stanford scheme. Finally, the recognizer will generate NE recognized Amharic text and send it as an output.

4.1.5 Performance evaluation

As we have discussed under Corpus preparation for ANER, for both phases (training and testing) 13,538 tokens were used. For training and testing we have

used, 12296 and 1242, respectively. 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).

$$P=\frac{T_P}{F_P+T_P}$$

And

$$R=\frac{T_P}{F_N+T_P}$$

Where, True Positives (T_P) are NE tags which are produced by the model that match the reference tagging , False Positives(F_P) are NE tags returned by the model that are not in the reference tagging.
And lastly, False Negatives (F_N) are NE tags in the reference tagging bur they are missed by the model.

The F measure which is computed by the uniformly weighted harmonic mean of precision and recall:

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

Performances that we have discussed in the following section are performed according to these performance measurements.

4.2 Experimentation Result

In this section we have discussed experiments which are conducted in different scenarios. The different scenarios have taken by varying our feature sets. The first section presents corpus used in the experiment then describing feature set in the experiment. In the second section experimentations with their performance achievement are presented. The final section discussed experiment results with their graphical representation.

4.2.1 Corpus size

The number of tokens used in this work are 13538, from these 10% of the total tokens are used for testing and the rest are supplied for model creating.

Table 4.1 corpus size

Number of tokens in training phase	Number of tokens in testing phase	Total number of tokens
12296	1242	13538

Feature sets plays a crucial role in CRF frameworks, and as we have stated under objective of the study, finding the optimal feature sets for Amharic language is our intended objective. To find these feature sets we have used word based features contains suffixes and prefixes of the current and surrounding tokens and context based features that are previous and next words of a particular word with their named entity tags.

- Suffixes- 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 four of the current and surrounding word have been taken for this language since it gave better performance than using length up to 2.

- Prefixes - A fixed length prefix of the current and the surrounding words might be treated as features, in this work, the tool we have used did not allow us to use prefix and suffixes separately.
- Previous word- words that occurred before to a current word and it might be also a clue word that indicates NE class of succeeding word.
- Next word- words that occurred next to a current word and it might be also a clue word that indicates NE class of preceding word.
- Named Entity Tags- the NE tags are used in our work as a feature, these features are used by increasing our window size up to 4.
- Orthographic – word shape of a token was one of our features set. The experiment also conducted by increasing the window size for orthographic feature and combining it with other features.
- Word pairs - features that are taken by calculating frequency co-occurrence of words.

The above listed features are used in this work, but before deciding feature set of this system determining the first feature set for our baseline experiment was the toughest task during experimentation. This raised due to a reason that the earlier ANER system feature set are not compatible with the tool we have chosen, Lingpipe which is a tool that has been used by previous ANER researchers. It accepts suffix and prefix independently, beginning of sentence (BOS) and End of a sentence (EOS) are also additional features in their tool. The other feature is Part Of Speech tagging (POS), the feature that we have not used. As we have discussed previously, Stanford features have a capability that POS can performs. According to Stanford NER developers group, the features used by POS tagger are very similar to those used in the NER system, so there is very little benefit to using POS tags[3].

With all the above mentioned factors, using previous work features as a baseline experiment were impossible. Hence, starting our experiment with all

[3] http://www-nlp.stanford.edu/software/crf-faq.shtml#pos

the same features that have been used in the previous work is difficult we have selected common features from them. But feature usages are still different.

For our baseline experiment we have selected the following features that have been used and brought the optimal performance in previous work.

Table 4.2 Baseline experiment

Experiments	Features used	Performance		
		Recall %	Precision%	F-measure %
Baseline	Window size (left and right)=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag	76.84	82.95	79.78

NB. The features that we removed from previous work are BOS, EOS and POS, performance achieved by features in the table was F-measure of 74.61 % and it was the highest performance in their work. This performance was achieved with 10405 total token sizes.

After we have selected baseline experiment features for our work, identifying optimal feature set for ANER is continued. In order to begin the first experiment, we have started by selecting a single feature with a single window size; this is because of the assumption that studying a role of each feature independently would help to come up with optimal feature sets. With this reason previous word feature with one window size has chosen for the first experiment.

Table 4.3 Experiment one

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
1.	Left Window size=1 Previous word	52.6	75.94	62.15

The second experiment is conducted by increasing the window size of previous experiment by one that is two. Window size increments added was on the left side only and F-measure increased by 0.20%.

Table 4.4 Experiment two

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
2	Left Window size=2 Previous word	52.6	76.51	62.35

From table 4.4 we understood that window size increment has performed better than a single window size so, the next experiment used bigger window size. The third experiment was conducted by taking next word along with the above window size. The F- measure is the same as experiment 2 (i.e. previous word with window size two).

Table 4.5 Experiment three

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
3	Left Window size=2 Next word	52.6	76.51	62.35

After analyzing the influence of next and previous word feature independently the next step was studying the role of these two features combination. Combination of two features is taken with a window size of two to the left side.

Table 4.6 Experiment four

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
4	Left window size=2 Previous word Next word	52.33	75.94	61.97

As it is shown in Table 4.6 F- measure is reduced by 0.38% from F-measures achieved when features applied independently. Even though the F- measure is reduced, using previous and next word in combination has its own advantage. F-measure was not the only decreasing performance recorded but also Recall and precision too. Based on experiment 4 additional features are selected. Now, the identified features which positively recognized NEs are two window sizes to the left, previous and next words. The next experiment was conducted by using suffix and prefix with a length of two and four. As we have discussed in section 2.7.4.3**,** this work used transliterated news and most Amharic scripts are represented by two English alphabets. With this assumptions our instant prefix and suffix length started from two and finally reached to four.

Table 4.7 Experiment 5 and 6

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
5	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 2	63.87	75.31	69.12
6	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4	65.46	76.05	70.36

In table 4.7 the F- measures of our system has increased on both experiments, experiment 5 and 6 by 6.77 and 8.21, respectively. Then, based on the above result we have preceded to the next experiment by adding previous and next NE tags independently and combination of them to experiment 6. From our previous experiment (i.e. experiment 6) results in experiment 7, 8 and 9 shows, F- measure has increased by 9.42, 7.69 and 9.97, respectively.

Table 4.8 Experiment 7, 8 and 9

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
7	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag	76.84	82.95	79.78
8	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Next word tag	76.19	80	78.05
9	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag	75.52	85.8	80.33

Here in experiment 9, the F- measure the system scored increases the baseline experiment by 0.55%. The next experiment has taken all the features in experiment 9 with a new window size. In experiment 10 we took window sizes of two, 1 to the left and right side of the current word.

Table 4.9 Experiment 10

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
10	Window size (left and right)=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 2 Previous word tag Next word tag	76.84	84.88	80.66

The F-measure of experiment 10 is increased by 0.33 % from our previous experiment result. The following experiment is conducted to show the role of other additional features. First we took all features from experiment 9 then, add orthographical and word pair features independently, finally combination of these two features has also been tested as follows:

Table 4.10 Experiment 11, 12 and 13

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
11	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape	75.52	85.8	80.33

12	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word Pairs	74.48	86.67	80.12
13	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape Word pairs	74.1	86.67	79.9

The above three experiments have shown that , orthographical features brought the same F- measure , Precision and Recall as features which have been used in experiment 9. Whereas , experiment 12 which is addition of word pairs feature only and experiment 13 , combination of word pair and orthographical feature have scored less F- measure performances than the previous experiment. During experiment 12 and 13 the F- measure recorded was reduced by 0.21% and 0.43 %, respectively.

The final experiment that we have conducted has taken all features in experiment 13 with one additional window size to the left. Here in experiment 14 F- measure has increased by 0.1% from experiment 13.

Table 4.11 Experiment 14

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
14	Left Window size= 2 Right window size=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape Word pair	76.84	83.43	80

4.2.2 Discussion of Experiment results

The above experiments show that different combination of features set have brought various F- measures, Recall and Precision. The following table presents performance of the systems.

Table 4.12 Performance

Experiments	Performance		
	Recall %	Precision%	F-measure %
Baseline	76.84	82.95	79.78
1	52.6	75.94	62.15
2	52.6	76.51	62.35
3	52.6	76.51	62.35
4	52.33	75.94	61.97
5	63.87	75.31	69.12
6	65.46	76.05	70.36
7	76.84	82.95	79.78
8	76.19	80	78.05
9	75.52	85.8	80.33
10	76.84	84.88	**80.66**
11	75.52	85.8	80.33

12	74.48	86.67	80.12
13	74.1	86.67	79.9
14	76.84	83.43	80

From baseline to the last experiment that is experiment 14, removing and increasing the number of features sets and window size have been taken placed. Removing and adding of features are followed researchers best rationality.

Features we have used in the baseline experiment are selected by the method that we have mentioned above.

To select the first experiment features we started from the baseline experiment and study the role of each feature in ANER. With this assumption analyzing the result of previous and next word features are conducted by changing the window sizes.

By using of previous and next words independently with the same window size we have achieved the same precision, recall and F- measure. After studying the performance from experiment 1, 2 and 3, the features identified for the next level were previous word, next word and window size of two from the left side of a current word. The main finding here is, a different window size have changed the performance in each steps and this clearly shows previous and next word features independently with a window size of two have performed better than a single window size.

Based on this understanding experiment 4 is conducted and it achieved less performance. This experiment shows, a combination of previous and next word with different window sizes degrades performance of ANER system. After conducting different combination of the above three feature sets, looking for the next feature set is continued by using all features that have been used in experiment 4.

In experiment 5 and 6, the length of prefix and suffix we took shows a great change from previous experiments. By varying lengths of these two features are also indicated there is a change in the performance. From these two experiments we have observed that prefix and suffix could be one of the optimal

features in ANER system. Then, by using features in experiment 6 we continued for further optimal features and studying the role of previous and next word tags of a current token were taken placed.
Here in the following experiments we have followed the same procedure as previously conducted experiments. From the three experiments that take each feature independently and their combination shows a tremendous change from experiment 6. As we have observed in the experiment, the combination of previous and next tag performs better than performances that were recorded when they are added independently. But previous tag performs better than next word tag.

After experiment 9, we have gone back to the baseline experiment, this is because of in experiment 9 combination of previous and next tag outperforms on applying previous tag only. By taking the result of experiment 9 into consideration, experiment 10 was conducted by adding next tag to features in the baseline experiment and we got the highest performance of the entire experiments.

Finally , based on the performance achieved in experiment 9 finding other optimal features are supported by adding orthographical feature, word pair feature and combination of them with different window size were conducted. Performance achieved in four of the above experiments shows degradation from the previous one and the researchers reached to, a simple addition of many features could not improved the performance of ANER system.

The general graphical representation of the above experimentation discussion is presented by clustering our fifteen experiments in to five clusters. Clustering is based on feature set which have been used in the experiments. The first cluster contains experiment 1, 2, 3 and 4; those are conducted by using previous and next word features with different window size. Experiment 5 and 6 are grouped in to second scenario that clearly shows performance changes over prefix and

suffix features with different length. Previous word tag and next word tag used in experiment 7, 8 and 9 are clustered in to the third scenario. The fourth cluster (experiment 11, 12, 13 and 14), holds word pair and word shape feature sets with different window size and finally baseline experiment and our highest performance experiment (i.e. 10) are fallen in to cluster 5.

Based on the above clustering , first the change within each clusters are presented then the second graphical representation held five experiment results by taking transition performance, which is an experiment performance that we added a feature to precede for the next experiment. In the second type of graphical representation the researchers took experiment 14 from the fourth cluster, this is because there was no transition after this experiment and this is one of the experiment that used all feature set of ANER system.

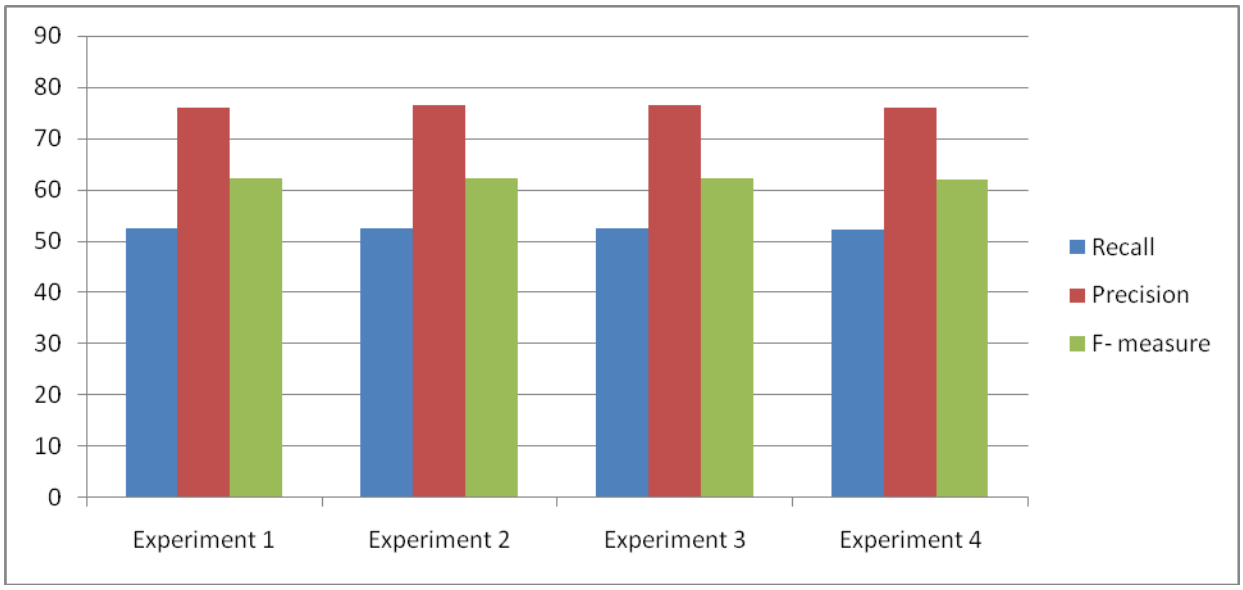

Figure 4.2 Scenario 1.

The first scenario shows F- measure did not increase when we change window sizes of previous and next word features .

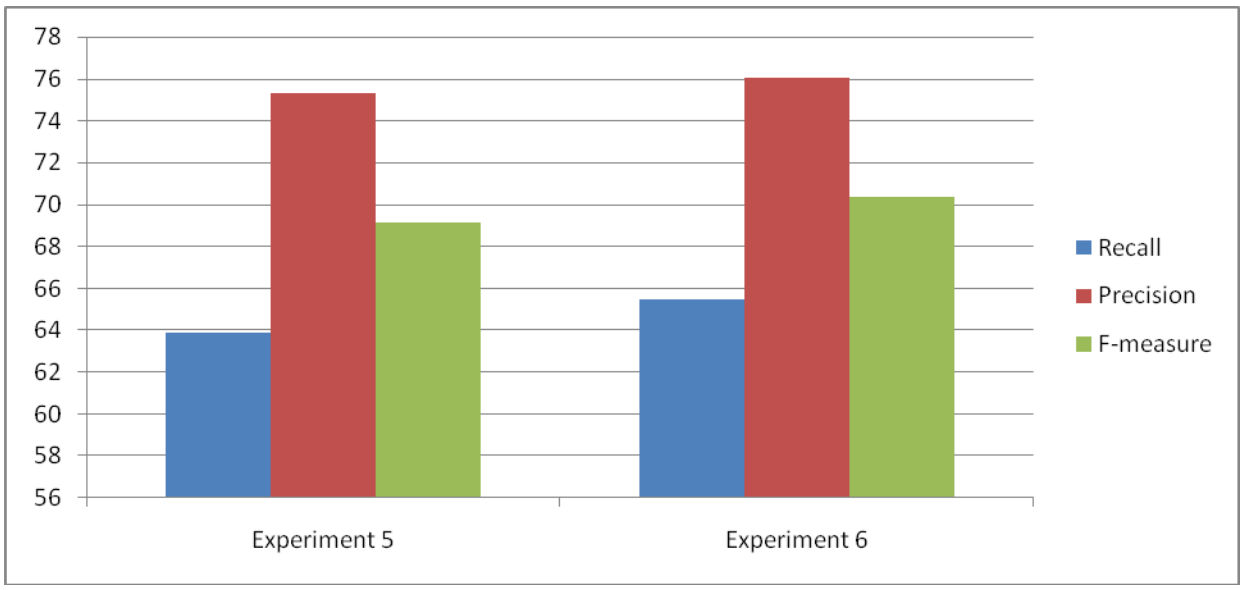

Figure 4.3 Scenario 2

It is our first obsrvation that , prefix and suffix have plyed a role in identifying Amharic NEs. as it is shown in the aove graph within each experiments suffix and prefix length also alter NE recognizing ability of a system.

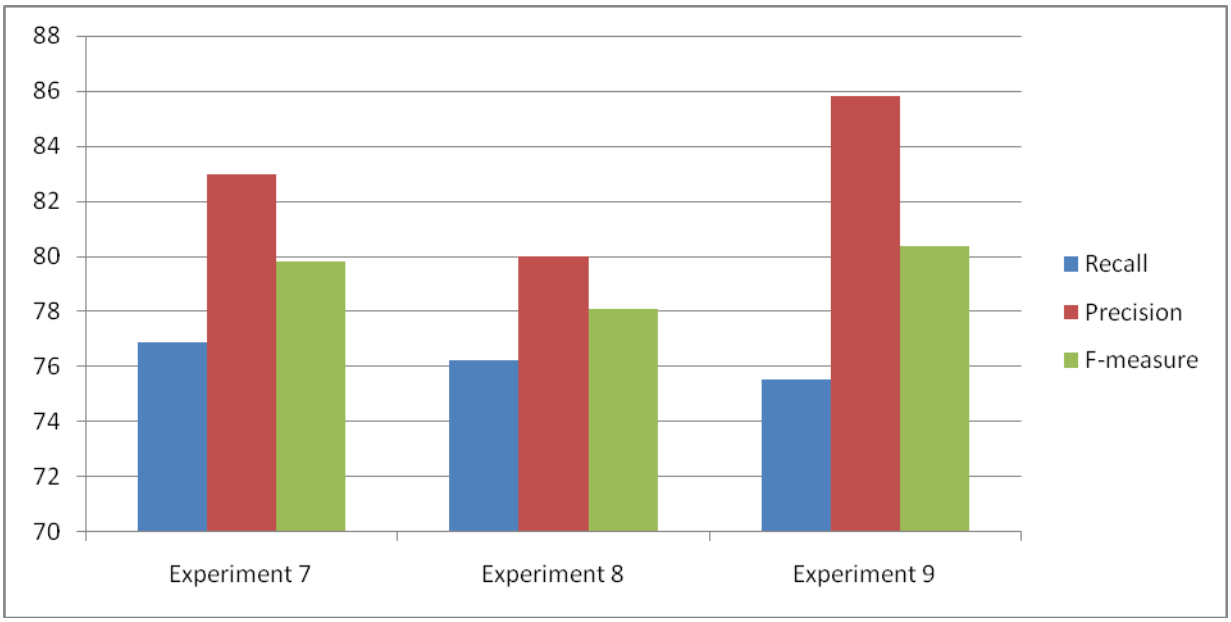

Figure 4.4 Scenario 3

After prefix and suffix , the drastic change has been obsrved during previous and next tag features. the change in each feature has changed . The worst perfrmance during scenario 3 is independent usage of next word feature and the maximum was combination of the two features. the precision in these scenario is one of the greatest from previous experiments.

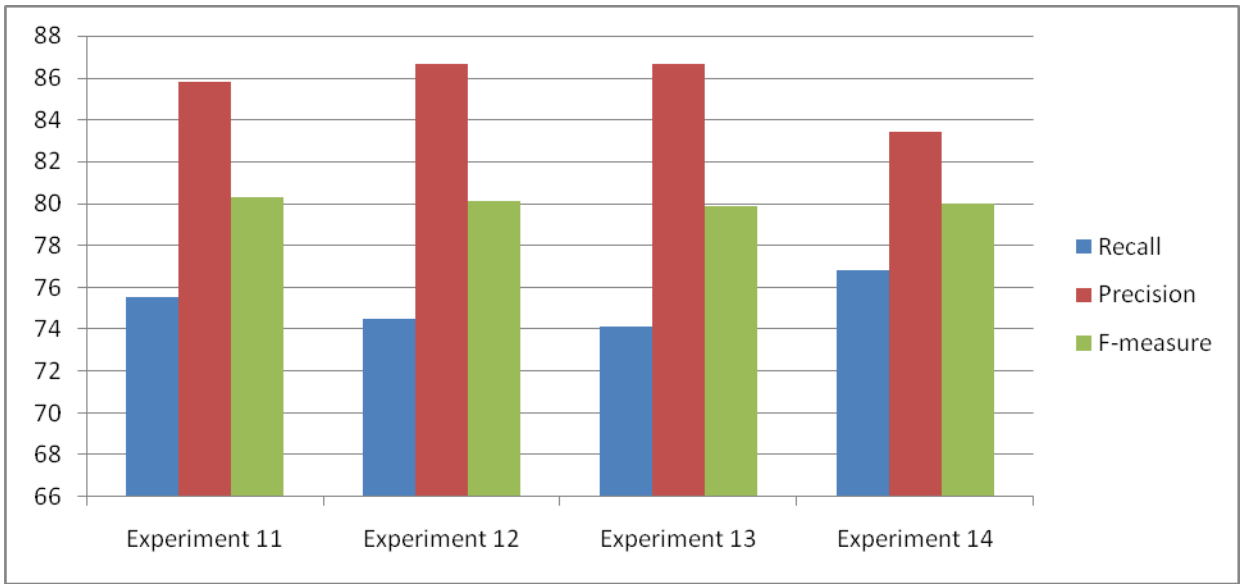

Figure 4.5 Scenario 4

As compared to scenarion3 the change in scenario 4 is insignificant . Recall and precision in scenario 4 are varied but F- measure in experiment 11 repeats experiment 9 result, where as the rest experiments performed with less F-measure that experiment 11 scored. The performance in scenario 4 indicated after certain incriment of feature set the performace will saturate and with worst case it inclined to reducing F- measure of the system . the reason behind this performance reduction need further study.

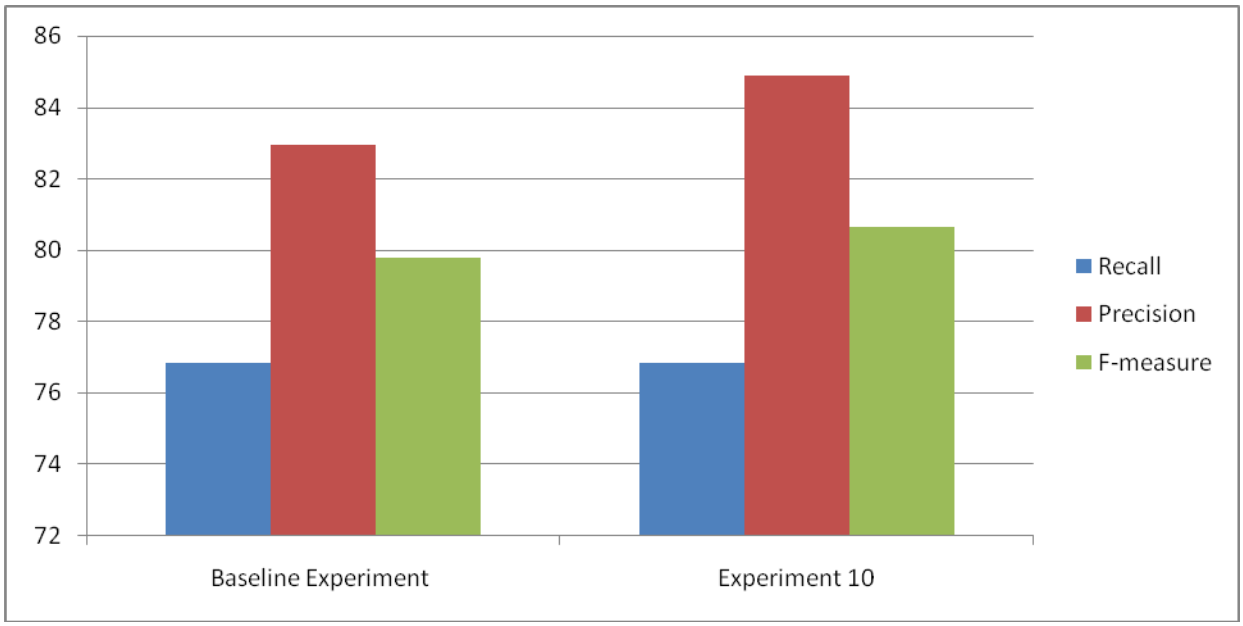

Figure 4.6 Scenario 5

The final cluster in the first graphical representation is scenario 5. In this scenario precision, recall and F- measure changes from baseline to experimt 10, as it is mentioned in scenario 3 combinational usage of previous and next tags have a potential to recognize Amharic NEs and the new finding in this scenario is , performace of a system varied in combinational usage of previous and next tags with different window sizes.

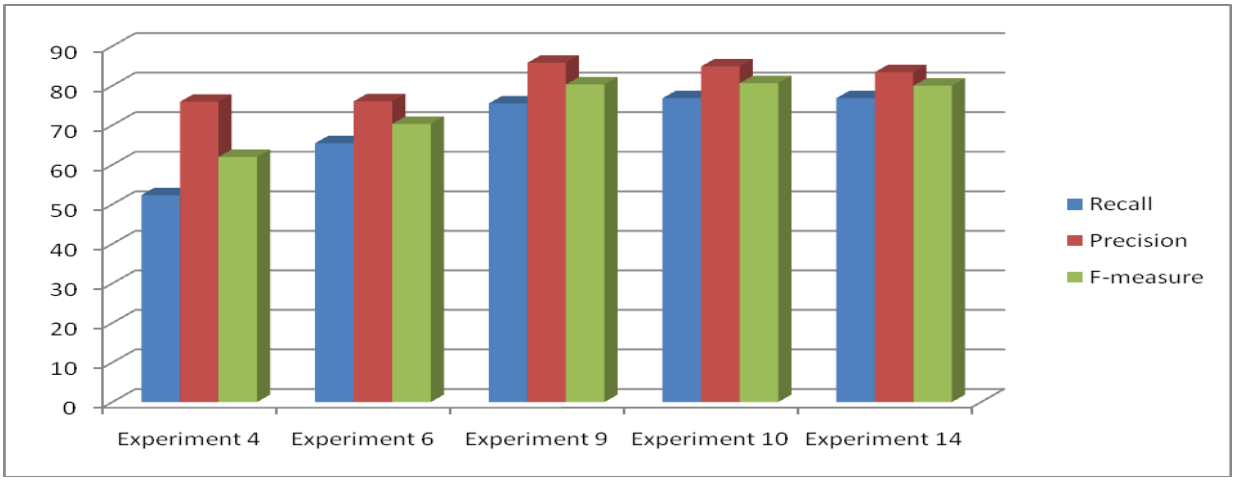

Figure 4.7 Second scenario graphical representation

From the above graph we understood that, Recall, Precision and F- measure of the system have been changing from experiment 4 to 6 and 9 in a tremendous way then it increased with a little decimal value in experiment 10. Finally the performance of the system reduced in experiment 14.

Generally, during our experimentation the identified optimal features that bring tremendous change are : suffix and prefix with 4 word length , previous and next word tag of a current word and window size of features.

Chapter five

Conclusion and recommendation

5.1 Conclusion

Named entity recognition is the process of identifying the right tag for names in a document. Many NER systems are developed for English and other European languages, next to these, NER for Arabic and south Asian languages have been developing. Amharic is one of the languages that have many speakers in Ethiopia, but it is under resourced.

This work tried to contribute one important point which plays a role for overcoming some challenges in NLP regarding to Amharic. Having a good NER system could be used in information extraction of any domain specific tasks, question answering, information retrieval … etc. to come up with a system that could be used in the above application areas, identifying optimal feature set is the most important task of any NER. This work is also delimited to identifying these features to recognize PERSON, ORGANIZATION and LOCATION names of transliterated Amharic corpus.

This study has proposed and designed a system called Amharic Named Entity Recognition. The system is designed based on a supervised machine learning component. We implemented the machine learning component using CRFs algorithm. To do this research, 13538 tokens were taken from WIC corpus and tagged manually. These data sets are classified into training and testing. By using the data processes of identifying optimal features for ANER have been implemented, we have conducted fifteen experiments and the roles of each feature in different experiments were tested. The highest performance achieved in this work is 80.66%, with a window size of two on both sides, previous and next tag of a current word and prefix and suffix with length four. The worst performance achieved is 61.97%, feature sets are a window size of two from the left side of a word and previous and next word of the current token.

The previous work on ANER has performed with 74.61 on a different tool, corpus size and feature set. Due to these reasons we could not compare this work with the previous one. But from the finding we got, a tool might have impact on the performance of ANERs.

From the findings, prefix and suffix with a length of four, previous and next NE tag of a token and window size features are the observed optimal feature sets for recognizing Amharic named entities.
During combining a feature set one needs to be careful in grouping the features; otherwise it degrades system's predicting ability.

5.2 Recommendation

This study showed some of the features that have never got an attention of previous ANER. The involvement of those new features has brought a promising performance and the following points are some of our observation that should be taken in consideration for future works of ANER.

- Due to the limitation of the tool that is used for this study, we could not observe the separate role of prefix and suffix. So, separating these two features and identifying whether their combinations or independent role should be included in ANER optimal feature set is recommended.
- Experimentation with regard to corpus size has never been conducted in this work and the corpus size in this work is very small with regard to CRF based NER systems and training by different corpus size would give a better performance
- From literatures we have reviewed[7, 8] used Lingpipe and Stanford, respectively and performance achieved with different corpus size and features sets were 76.6 and 84.71, respectively , these shows further research regarding to NER tool needs to be.

- Developing a hybrid approach which incorporates a rule based on the nature and pattern of Amharic NEs with CRF based approach is recommended since it might give a better performance.

References

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes,* vol. 30, pp. 3-26, 2007.

[2] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 142-147.

[3] S. M. Algahtani, "Arabic Named Entity Recognition: A Corpus-Based Study," 2012.

[4] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1-7.

[5] A. A. Argaw and L. Asker, "An Amharic stemmer: Reducing words to their citation forms," in *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, 2007, pp. 104-110.

[6] M. A. Mehamed, "Named Entity Recognition For Amharic Language," Msc., Computer Science, Addis Ababa University, Addis Ababa, 2010.

[7] H. Keno, "DESIGNING A NAMED ENTITY RECOGNITION SYSTEM FOR AFAAN OROMO TEXT," MSC, Information Science, Addis Ababa University, Addis Ababa, 2012.

[8] M. L. Kejela, "Named Entity Recognition for Afan Oromo," Computer Science, Addis Ababa University, Addis Ababa, 2010.

[9] S. S. Keerthi and S. Sundararajan, "CRF versus SVM-struct for sequence labeling," Technical report, Yahoo Research2007.

[10] D. Li*, et al.*, "Conditional random fields and support vector machines for disorder named entity recognition in clinical texts," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, 2008, pp. 94-95.

[11] P. Thompson and C. Dozier, "Name searching and information retrieval," in *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, 1997, pp. 134-140.

[12] T. Sehgal*, et al.*, "NLP & its Applications."

[13] A. Mikheev*, et al.*, "Named entity recognition without gazetteers," in *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 1999, pp. 1-8.

[14] E. Riloff and J. Lorenzen, "Extraction-based text categorization: Generating domain-specific role relationships automatically," *Natural language information retrieval,* pp. 167-196, 1999.

[15] J. Cowie and W. Lehnert, "Information extraction," *Communications of the ACM,* vol. 39, pp. 80-91, 1996.

[16] M.-F. Moens, *Information extraction: algorithms and prospects in a retrieval context* vol. 21: Springer, 2006.

[17] K. Kaur and V. Gupta, "Name Entity Recognition for Punjabi Language," *Machine translation,* vol. 2, 2012.

[18] R. Farkas, *et al.*, "The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning---Shared Task*, 2010, pp. 1-12.
[19] A. Kao and S. R. Poteet, *Natural language processing and text mining*: Springer, 2007.
[20] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning,* vol. 3, pp. 1-130, 2009.
[21] O. Chapelle, *et al.*, *Semi-supervised learning* vol. 2: MIT press Cambridge, 2006.
[22] D. Nadeau, "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision," Phd, University of Owtawa, Canada, 2007.
[23] E. Alpaydin, *Introduction to machine learning*: The MIT Press, 2004.
[24] J. Mayfield, *et al.*, "Named entity recognition using hundreds of thousands of features," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 184-187.
[25] M. Tkachenko, *et al.*, "Named entity recognition: Exploring features," in *Proceedings of KONVENS*, 2012, pp. 118-127.
[26] Y. Benajiba, *et al.*, "Arabic named entity recognition using optimized feature sets," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 284-293.
[27] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 2009, pp. 147-155.
[28] A. Ekbal and S. Bandyopadhyay, "Bengali named entity recognition using support vector machine," in *Proceedings of workshop on NER for south and south east Asian languages, 3rd international joint conference on natural languge processing (IJCNLP)*, 2008, pp. 51-58.
[29] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 2003, pp. 134-141.
[30] R. Klinger and K. Tomanek, *Classical probabilistic models and conditional random fields*: TU, Algorithm Engineering, 2007.
[31] J. Lafferty, *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.
[32] B. Sun, "Named entity recognition: Evaluation of Existing Systems," Norwegian University of Science and Technology, 2010.
[33] M. Marrero, "Advances in Computational Linguistics," *Research in Computing Science,* vol. 41,2009, pp. 47-58, 2009.
[34] C. W. Isenberg, *Grammar of The Amharic Language*. London: Richard Watts 1976.
[35] B. M. HAILEMARIAM, "N-gram-Based Automatic Indexing for Amharic Text," Msc, SCHOOL OF INFORMATION STUDIES FOR AFRICA, Addis Ababa University, Addis Ababa, 2002.
[36] G. A. Demeke and M. Getachew, "Manual annotation of Amharic news items with part-of-speech tags and its challenges," *Ethiopian Languages Research Center Working Papers,* vol. 2, pp. 1-16, 2006.
[37] A. A. Argaw, "AUTOMATIC SENTENCE PARSING FOR AMHARIC TEXT AN EXPERIMENT USING PROBABILISTIC CONTEXT FREE GRAMMARS," Msc, Computer Science, Addis Ababa University, Addis Ababa, 2002.

[38] L. Bender and C. Ferguson, "The Ethiopian writing system," *Language in Ethiopia. Edited by ML Bender, JD Bowen, RL Cooper, and CA Ferguson. London: Oxford University press*, 1976.
[39] T. Zhang and D. Johnson, "A robust risk minimization based named entity recognition system," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 204-207.
[40] N. Jahan*, et al.*, "Named Entity Recognition in Indian Languages Using Gazetteer Method and Hidden Markov Model: A Hybrid Approach."

Appendixes

Appendix 1. The Amharic character set [38]

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					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ᎀ	ᎁ	ᎂ	ᎃ	ᎄ	ᎅ	ᎆ

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

A Named Entity Recognition for Amharic

By

Besufikad Alemu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE:

Addis Ababa, Ethiopia

June, 2013

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

A Named Entity Recognition for Amharic

By

Besufikad Alemu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE:

Addis Ababa , Ethiopia

June, 2013

DECLARATION

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

_____

Besufikad Alemu

June, 2013

Thesis has been submitted for examination with our approval as University advisor.

_____

Solomon Teferra (PHD)

June, 2013

Acknowledgement

First and foremost, I would like to thank God, who makes everything possible. I would extend my sincere gratitude to my advisor Dr. Solomon Teferra for his critical comments and for being my driving force throughout this work.

My special thanks goes to my sister, Yilfashewa (Beti), for making all those hardships I had to go through easier. Beti, you are the greatest sister, mom and best friend. I could not have been able to make it without you.

I would also thank Moges Ahmed and my friends for their understanding and support.

Contents

List of table

List of figure

Abbreviation

MUC :	Message Understanding Conference
IE :	Information Extraction
NERC:	Named Entity Recognition and Classification
ORG:	Organizations name
PER :	Persons name
LOC :	Locations name
CONLL :	Conference on Computational Natural Language Learning
NER :	Named Entity Recognition
POS:	Part of Speech tag
NERs:	Named Entity Recognition System
NLP :	Natural Language Processing
ANER :	Amharic Named Entity Recognition
CRF :	Conditional Random Fields
IE :	Information Extraction
U.S.:	United States
SL :	Supervised Learning
HMM :	Hidden Markov Models
ME :	Maximum Entropy
SVM :	Support Vector Machines
UL :	unsupervised learning
NLTK :	Natural language tool kit
EM :	Expectation Maximization
SSL :	Semi- supervised Learning

Abstract

This thesis describes the development of Named Entity Recognition (NER) system for Amharic. 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 work is conducted by using machine learning method, Conditional Random Fields (CRF). Amharic Named Entity Recognition (ANER) is proposed based on a supervised learning which is CRF machine learning. Amharic corpus of size 13,538 words have been developed with Stanford tagging scheme.

Since the objective of the work is working on feature set of Amharic Named Entities (NE), fifteen experiments were conducted by interchanging different features. Previous and next word, named entity tag of a word, word pairs, word shape, prefix and suffix features have been used to identify three NEs, Person, Organization and Location.

The results of the experiment show that proper features combination in NER has a great role. The highest F-measure achieved in this work is 80.66%, with a window size of two on both(left and right)sides, previous and next tag of a current word and prefix and suffix with length four. The worst performance achieved is 61.97%, feature sets are a window size of two from the left side of a word and previous and next word of the current token. Finally the identified optimal feature sets from the experiment are: prefix and suffix with a length of four, previous and next NE tag of a token and window size features

Key word: Named Entity Recognition, Conditional Random Fields, Amharic Named Entity Recognition, Named Entities

Chapter One

Introduction

1.1. Background

According to (Grishman and Sundheim ,1996) as they are cited on [1], the term "Named Entity", now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC-6). At that time, MUC was focusing on Information Extraction (IE) tasks where structured information of company activities and defense related activities is extracted from unstructured text, such as newspaper articles. In defining the task, people noticed that it is essential to recognize information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions. Identifying references to these entities in text was recognized as one of the important sub-tasks of IE and was called "Named Entity Recognition and Classification (NERC)" [1].

Named entities are phrases that contain the names of persons, organizations and locations. Example:
[ORG AAU] Official [PER Besufikad] heads for [LOC Addis Ababa] .
This sentence contains three named entities: Besufikad is a person, AAU is an organization and Addis Ababa is a location. Named entity recognition is an important task of information extraction systems. There has been a lot of work on named entity recognition, especially for English [2].
According to [1], a good proportion of work in NER research is devoted to the study of English but a possibly larger proportion addresses language independence and multilingualism problems. German is well studied in CONLL-2003 and in earlier works. Similarly, Spanish and Dutch are strongly represented, boosted by a major devoted conference: CONLL-2002. Japanese has been studied in the MUC-6 conference, in addition to the above mentioned

languages he has listed many European languages that have been tried to develop a good NER.

1.2. The challenge of Named Entity Recognition and Possible Solutions

The ambiguity of NEs in text resides at two different levels; detection and classification. The first involves disambiguating NEs from non NEs in text and the second involves classifying them into classes e.g. person or location. The two processes of NER are sequential and the correctness of the identification phase is a prerequisite for the classification phase. The second level is dependent on the first one; NEs will not be classified correctly if they are not detected correctly in the first place [3].

The level of ambiguity that resides at these two levels differs from one language or domain to another. In English text, for instance, the identification would normally rely on orthographic case information (lower and upper) to detect NEs. If we manage to identify all NEs using that information then the fundamental problem is classifying them. For the classification, assuming that we managed to collect and group all NEs into lists for each NE class, there still would be cases where one token might fall into more than one list [3]. For instance, in Amharic the word *Amede* could be a person name or location. According to [3], Thus more information is required to classify NEs correctly; for example, the context in which the word occurs, such as the previous word. Thus, if the word *Amede* is preceded by *Ato* then it is more likely to be a person name. Titles and designators have proved successful in NER and are called triggering words or triggers.

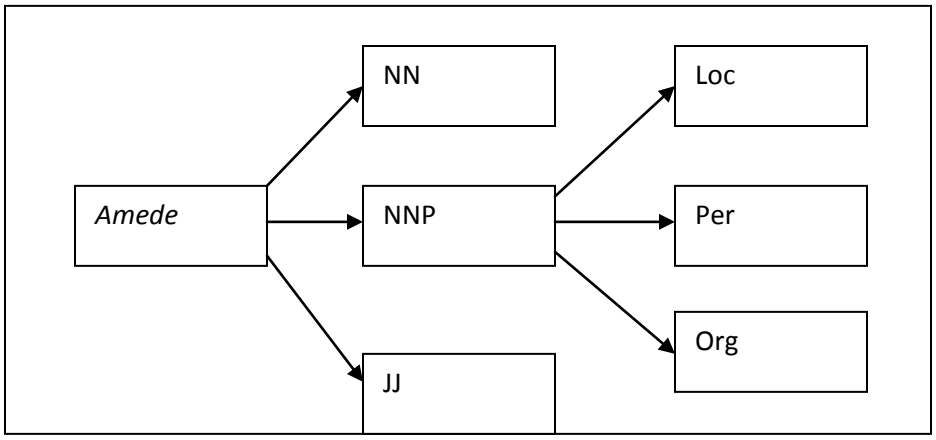

Figure 1.1 State- of- a sentence ambiguity example Shabib (2011) [1]

To be able to identify NEs successfully in such a situation, there would be a need to analyze the context, given that each word category would have a different context. For instance, an adjective typically cannot follow a verb without a determiner and also noun typically cannot be followed by an adjective in English. Thus, analyzing the surrounding lexical and POS context is crucial. If we have the means to generate the POS information correctly, then the problem would be solved to a large extent. However, it is not feasible as POS taggers rely on case information to tag proper nouns [3].

[1] proper noun (NNP), noun(NN), adjective (JJ), Location (L), person name(per), and organization (Org)

1.3 Statement of the problem

The motivation that initiated the researcher to conduct this research on NER is the advantageous and application areas of NERs. As it is stated in [4], NER is a key technology of Information Extraction and Open-Domain Question Answering.

According to [5] , Amharic 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 is used nationwide as the official language of the nation. Despite having large speaker population, the language has little computational linguistic resources. So, conducting a research on such kind of language is advisable to overcome the current shortage of computational linguistic resources.

Amharic is one of the under-resourced languages. Hence, there is no full-fledged Amharic Named Entity Recognition system that could contribute towards designing NLP. So that conducting Amharic NER with best performance will have a contribution to advance research in NLP for Amharic language.

There is only one attempt on Amharic named entity recognition. And further improvement regarding Amharic NER(ANER) has never been tried after the first ANER [6] attempt. Nationwide also there have been limited attempts for developing a NER of Ethiopian languages, Amharic by Moges[6] , Afan Oromo by Habtamu [7] and Mandefro [8] were attempted to develop Named Entity Recognitions. Moges [6] shown that some features played a great role in changing the performance of the Amharic NER system. So, in the proposed system the researcher tried to reconsider features that could increase the accuracy of Amharic NER.

According to [9] the best choice of feature combination chosen via the validation set correlated perfectly with the choice that has the best test set

performance, and in a system like [6] which followed Conditional Random Field (CRF) model, well-chosen features along with dictionary-based features tend to improve the CRF model's performance [10]**.** These shows considering other features would help to achieve a better performance. Therefore it is the present research to explore best feature sets for designing a high performance Amharic Named Entity Recognition.

Hence the research questions to be addressed have been the following

- What are the best feature sets for enhancing the performance of Amharic NER?
- Does a combination of different feature have an effect on the system?
- Which features are the optimal features sets in ANER?

1.4 Objectives of the study

General and specific objectives that have been tried to achieved are:

1.4.1 General objective

The general objective of this research is identifying the optimal feature sets that improve the performance of ANER system.

1.4.2 Specific objective

- Review literature so as to understand the role of named entity and approaches in NER
- Identify as many features as we can.
- Identify the optimal combination of the features that brings the best ANER Performance
- Selecting features that could be integrated with Amharic NER.
- Compare models that could help to improve the performance of Amharic NER.

- Evaluate performance of the proposed system by using a test dataset

1 .5 Significance of the study

NER was successfully incorporated into name searching systems to identify names in queries and the underlying data source for information retrieval [11]. Another important application is question answering, which was driven by one of Text Retrieval Conference's tracks. According to (Noguera et al. 2005) as cited by [3] NER is a core component of these systems and the effect of NER was measured in. The idea was to consider only documents that have at least one named entity provided in the query. It was found that NER could reduce the data returned by the IR system by 62% without any loss of information and by 92% with acceptable loss. Detecting NEs in a query was also success [3].
Conducting NER for Amharic is also very important for the development of Natural Language Processing (NLP) tools (such as, Speech synthesis, speech recognition, Question Answering, Information retrieval, Information extraction) [12]. In order to achieve such kind of advantages, developing a system with a high performance is unquestionable. In order to have a high performance ANER, finding the optimal features and identifying the best features combination are the main task of NERs.

Result of this research could be an input for different disciplines of NLP specifically in Amharic. This includes, Information extraction, Question answering, Ontology construction etc.

1.6 Scope of the study

Most common among marked categories are names of people, organizations and locations as well as temporal and numeric expression [13]. And within each category there could be additional sub categories. For example, for location may have city, mountain, river, sea, village etc. as its subcategory. In

this study due to time constraint only the three main NE classes names of people, organization and locations without their subcategories were taken.

1.7 Methodology

NER is a compound task that is to be done by different components at different steps. The research also has different developmental stages. First finding out the corpus from a news source was researchers' initial task. The learning methodology to be adopted requires customization, so it has its own computational task. Selecting the appropriate tools and techniques were the parallel task during all the stages.

1.7.1 Literature review

This work is conducted by reviewing a number of related literatures from books, journal articles, conference proceeding and the internet, so that choosing appropriate tools and methods were supervised. The large portions of reviewed materials are conference and journal articles.

1.7.2 Tools

A tokenizer during preprocessing of both training and testing data was developed by using Python programming languages. The reason why the researchers choose python was, it is easy for text processing. For NER part, Stanford NER tool is used.

As [12] describes machine learning techniques which can handle multiple features, a machine learning approach was analysed as an approach for NE learning process.

1.7.3 Data source and preparation

As it is mentioned above the tool was tested with freely available Amharic texts, which are collected from WALTA. Due to the availability of transliterated articles and acquiring those articles for freely, WALTA is chosen.

The collected articles were further edited manually by removing those sentences which does not have NEs at all. The resulting data is then tokenized and tagged manually in accordance with the Stanford standard. Finally, NE corpus of size 13538 was developed.

1.7.4 Test Procedure

The performance of methods and tools are measured based on the three main scoring approaches: precision, recall and F-measure. According to [13], These measures are the most commonly used for the assessment of statistical extraction systems and trace their origins back to the information retrieval discipline.

1.8 Organization of the work

The rest of this thesis is organized as follows. Chapter 2 , literature review , concept related to named entity recognition is discussed. In the chapter background information of information extraction , feature sets types, machine learning approaches, CRF model and nature of Amharic language are described. In chapter 3 local and global literatures are reviewed and their achievements are discussed in. chapter 4 contains design and implementation part of this work, experimentations that are conducted in the research are found in this chapter. Finally, conclusion and recommendations which are given based on the experimentation are found in chapter 5.

Chapter two

Literature review

2.1 Information Extraction (IE)

Currently, there is a considerable interest in using these technologies for information retrieval, since there is an increasing need to localize precise information in documents, for instance, as the answer to a question, rather than retrieving the entire document or a list of documents.

According to [14],IE systems extract domain-specific information from natural language text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies, and physical objects. Domain-specific extraction patterns (or something similar) are used to identify relevant information

This early definition on information extraction is too limited and this definition is the same as current named entity extraction definition, Information extraction system should not necessarily be domain specific rather it should be domain independent or at least flexible to any domain with a minimum amount of engineering effort.

Some scholars are saying information extraction only extracts a fragment information but other says it is the process of extracting fragment information from a corpus and then process it in a way that it gives meaning. According to [15],IE isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework. The goal of information extraction research is to build systems that find and link relevant information while ignoring extraneous and irrelevant information.

As [16] defines IE is the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making the information more suitable for information processing tasks.

2.2 Named entity recognition

For the first time the name NER system coined at MUC-6 but the system was applied in different names those are domain specific (extract and recognize company names) . The term NER came after the increasing need of extracting specific information from a large corpus. To conduct a NER system defining the right tag for an entity is the most important task, there are conventions those defined tagging techniques of a named entity in a corpus. Message Understanding Conferences-6(MUC-6) and MUC-7 and Conference on Computational Natural Language Learning (CoNLL) are among the conventions that most researchers who are conducting on NER are currently using [17].

2.2.1 Message Understanding Conferences (MUC)

The Message Understanding Conferences (MUC) was a series of conferences organized by the (U.S.) National Institute of Standards and Technology and the U.S. Department of Defense Advanced Research Projects Agency. These events were held between 1987 and 1998 as part of the TIPSTER Text program [3] . It is believed that MUC seven conferences on IE contributed a lot for the development of NEP research particularly in IE. The MUC conferences focused on English, Chinese, Japanese and Spanish languages the three Identified MUC conference entities are ENMAX, TIMEX and NUMEX.

1. **ENMAX** contains, **Person name** - named person or family. Eg. Besufikad ,Alemu etc.

   Location name - name of politically or geographically defined location. Eg, Addis Ababa, Ethiopia etc.

   Organization name - named corporate governmental or other organizational entity. Eg. Addis Ababa University, International Leadership Institution etc.

2. **TIMEX** contains, **DATE** includes complete or partial date expression. ሃሙስ ጠዋት/Thursday Morning/ መጋቢት ፲፫ / March 22/ etc.

   TIME includes complete or partial expression of time of day ቅዳሜ ማታ / Saturday night/

3. **NUMEX** contains- numeric expressions, monetary expressions and percentages; expressed in either numeric or alphabetic form.

   Money- monetary expression eg. 5 birr, $3 etc.

   Percent – percentage expression eg. 5%, 100% etc.

From the above listed MUC conference entity types researchers conducted by using either partially or complete availability of the types.

2.2.2 Conference on Computational Natural Language Learning

CoNLL is highly focused on the named entity recognition task. The named entity types classification somehow differs from MUC, the three sub-classes in MUC first class (NAMEX) are treated as an independent class and by introducing Miscellaneous names which includes product names, names of services, etc [18]. CoNLL categorized named entity types in four groups. In contrast to MUC the languages considered by CoNLL were Spanish and Dutch in 2002 and German and English in 2003.

2.3 Machine Learning

Grobelnik M, Mladeni´c D (2005) as cited on [19]Different knowledge discovery methods have been adopted for the problem of semi-automated ontology construction including unsupervised, semi-supervised and supervised learning over a collection of text documents, using natural language processing to obtain semantic graph of a document, visualization of documents, information extraction to find relevant concepts, visualization of context of named entities in a document collection.

Machine learning algorithms can be classified as Supervised, Unsupervised and semi-supervised. The distinction comes from how the learner classifies data. In this section we will see the three kinds of machine learning.

2.3.1 Supervised learning (SL)

Supervised learning, the classes are predetermined. The classes are previously known. In a supervised learning, let the domain of instances be X, and the domain of labels be Y. Let P(x, y) be an unknown joint probability distribution on instances and labels X × Y. Given a training sample $\{(x_i, y_i)\}_{i=1}^{n} \overset{i.i.d.}{\sim} P(x, y)$, supervised learning trains a function f : X → Y in some function family F, with the goal that f (x) predicts the true label y on future data x, where $(x, y) \overset{i.i.d.}{\sim} P(x, y)$ as well [20].

According to [21]the goal of supervised learning is to learn a mapping from x to y, given a training set made of pairs (x_i, y_i). Here, the $y_i \in Y$ are called the labels or targets of the examples x_i. If the labels are numbers, $y = (y_i)^T{}_{i \in [n]}$ denotes the column vector of labels. Again, a standard requirement is that the pairs (x_i, y_i) are sampled i.i.d. from some distribution which ranges over X × Y.

The process in supervised learning is that a certain part of data will be labeled with known classifications. The machine learner's task is to search for patterns and construct mathematical models. These models then are evaluated on the

basis of their predictive capacity in relation to measures of variance in the data itself.

Supervised learning is the current dominant technique for addressing the NER. SL techniques include Hidden Markov Models (HMM), Decision Trees, Maximum Entropy Models (ME), Support Vector Machines (SVM) and Conditional Random Fields (CRF). These are all variants of the SL approach, which typically feature a system that reads a large annotated corpus, memorizes lists of entities, and creates disambiguation rules based on discriminative features [22].

2.3.2 Unsupervised Learning (UL)

Unsupervised learning algorithms work on a training sample with n instances $\{\mathbf{x}_i\}_{i=1}^{n}$. There is no teacher providing supervision as to how individual instances should be handled, Common unsupervised learning tasks include, clustering, where the goal is to separate the n instances into groups, novelty detection, which identifies the few instances that are very different from the majority and dimensionality reduction, which aims to represent each instance with a lower dimensional feature vector while maintaining key characteristics of the training sample [20].

In unsupervised learning, there is no such supervisor and we only have input data. The aim is to find the regularities in the input. There is a structure to the input space such that certain patterns occur more often than others, and we want to see what generally happens and what does not [23].

According to [23] the goal of unsupervised learning is to group data into clusters . In fact, the basic task of unsupervised learning is to develop classification labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group. These groups are termed clusters, and there is a whole family of clustering machine learning techniques.

One can try to gather NEs from clustered groups based on context similarity. There are also other unsupervised methods. Basically, the techniques rely on lexical resources (e.g., WordNet), on lexical patterns, and on statistics computed on a large unannotated corpus [22].

In unsupervised, the machine is not told how the texts are grouped. Its task is to arrive at some grouping of the data. Unsupervised learners are not provided with classifications.

2.3.3 Semi- supervised Learning (SSL)

Semi-supervised machine learning came up with the combination of supervised and unsupervised learning. As [21]defines, SSL is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information – but not necessarily for all examples. Often, this information will be the targets associated with some of the examples. In such kind of cases , the data set X = (xi)i∈[n]can be divided into two parts: the points Xl := (x1, . . . , xl), for which labels Yl := (y1, . . . , yl) are provided, and the points Xu := (xl+1, . . . , xl+u), the labels of which are not known, and he called this a standard semi- supervised learning.

According to [20],SSl is not a half way rather it is an extension of SL and SSl Also known as classification with labeled and unlabeled data (or partially labeled data), this is an extension to the supervised classification problem. The training data consists of both l labeled instances $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabeled instances $\{x_j\}_{j=l+1}^{l+u}$. One typically assumes that there is much more unlabeled data than labeled data, i.e.,$u >> l$. The goal of semi-supervised classification is to train a classifier f from both the labeled and unlabeled data, such that it is better than the supervised classifier trained on the labeled data alone [20].

2.4 Feature set 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. Authors supported the above idea with an 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 [7].

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”>”

But the above definition is working only for European language those are following capitalization to named entities (i.e. English), where as for other languages like Amharic, Hindu, Arabic …etc the above mentioned feature vector did not work.

One of the most challenging aspects in machine learning approaches to NLP problems is deciding on the optimal feature sets, there are different feature set and selecting the best features is not easy. In the baseline research (Moges 2010) the features have shown that a change and selection of them also was not easy. So, as [24] described when developing a NER for a specific language, having a large number of features at the beginning is advisable and it helps to come up with the optimal feature sets.

Different authors classified feature set in different ways, some grouped local knowledge and global knowledge [25]within each group there is also subgroups ,and others [24, 26] categorized the sub groups independently .

According to [25] local knowledge are feature which can be extracted from a token (word) being labeled and its surrounding context.

Token (word) – based features contains suffixes, prefixes and orthographic features (shape) of the current token. Context – based features are Previous and next words of a particular word. The performance we get from token based and context based features are different. They are also domain specific, as [25]describes context based feature outperform than word (token) in biomedical entities and in newswire the result is vice versa.

External knowledge features are additional features that determine the entity of a token beyond the context and word based feature. According to [25] External knowledge features are part-of-speech (POS) tags, shallow parsing and gazetteers.

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 [7]. Pos tagging by itself is challenging and time consuming due to this the baseline and other NERs are used manually tagged corpus and other[27] Because of the aforementioned reasons develops without it.

Phrasal clustering- is clustering a corpus in to n clusters by using different clustering techniques and use the phrase to identify tokens entity.

Gazetteers- list of predefined names of persons, locations, organizations etc. Even though developing gazetteers is not easy working on the available list is also difficult. Different problems are raised from the misuse of gazetteers. According to [28] semantic ambiguities are occurred when we use gazetteers, there is a situation that a single name could be person name , organization name or location name. For example, *Alem* can be name of a person, organization or location. In such kind of scenario it is difficult to decide the right entity for a token.

The other challenge of using a well prepared gazetteer is that they are memory intensive that comes from people names and organization names are different due to this there will be numerous names in the list but the occurrence of

those tokens in a corpus is rare sometimes they might not be presented once in the whole document and having those unused words in the corpus is wastage of our memory space. In addition to this a large coverage gazetteer would not guarantee a better accuracy. As [3]said this is due to the fact that person names are mostly found in form of nouns and adjectives. Whereas rearranging the list in the gazetteers could improve the performance, according to [13] rearranging means selecting the most important out of the predefined list and adding the remained names that believed to bring a better performance.

2.5 Conditional Random Fields (CRF)

Conditional Random Fields can be understood as the sequence version of Maximum Entropy Models, that means they are also discriminative models [29]. In addition to this, Conditional Random Fields are not tied to the linear-sequence structure but can be arbitrarily structured, in contrast to Hidden Markov Models. The Conditional Random Fields (CRF) are probabilistic models for computing the probability p (Y|X) of a possible output Y = (y_1,...y_n) ε Y^n given the input X = (x_1 ...x_n) ε X^n which is also called the observation. In this section CRF is discussed in detail that is used in our ANER model.

According to [30] linear chain CRFs is a special form of a CRF, which is structured as a linear chain that models the output variables as a sequence.

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

Where, j specifies the position in the input sequence x, but the weights λ_i are not dependent on the position j. This technique, known as parameter tying, is applied to ensure a specified set of variables to have the same value. n specifies the length of the sentence, m specifies the number of feature templates.

The normalization to [0; 1] is given by

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y} \in \mathcal{Y}} \exp\left(\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right) \right) .$$ 2.

As it is indicated in the above two equations, features are dependent on the label sequence and herewith on the state transitions in the finite state automaton. So it is important to point out that only a subset of all features f_i is used in every transition in the model.

According to [30],the strategy to build a linear-chain CRF summarized in the following way:

1. Construct Stochastic finite state automaton (SFSA) S= (S,T) out of the set of states S (with transition T=$(s, \dot{s}) \in S^2$). It can be fully connected but it is also possible to forbid some transitions because the decision made in this stage is depending on the training data and the transitions contained there.
2. Specify a set of features templates $F = \{g_1(\vec{x}, j), \ldots, g_h(\vec{x}, j)\}$ on the input sequence. This helps for the generating features f_i .

   g(x,i) $\{^{1}_{0}$ $^{\text{if } x_i = \text{ names}}_{\text{else}}$
3. Generate set of features $\mathcal{F} = \{\forall s, \dot{s} \in S.\ \forall g_o \in F : f_k(s, \dot{s}, g_o)\}$

There are two problems in a linear chain CRFs that plays a great role in varying the performance of NERs.

The first one is, there is a given observation x and a CRF M: here the problem is finding the most probable label sequence Y. This problem is the most common application of a conditional random field to find a label sequence for an observation.

The other one is, given label sequences Y and observation sequences X: How to find parameters of a CRF M to maximize P (Y|X, M) are also a problem. This problem is a question of how to train to adjust the parameters of M which are

especially the feature weights. In our system we have tried to address both of the above mentioned problems of linear chain CRF model.

2.5.1 CRF training

To train a CRF we need labeled data sequences. According to [31],CRFs training on unlabeled data is difficult. The labeled sequences are, { ($x^{(1)}$,$y^{(1)}$),…,($x^{(n)}$,$y^{(n)}$)}, where $x^{(1)}=x_{1:N1}$ the first observation sequence, and so on. The main objective for parameter learning is to maximize the conditional likelihood of the training data.

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

$$= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)\right)}{\sum_{\vec{y}'\in\mathcal{Y}}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right)} \right) \right] \quad 3.$$

During training there are features that are more than enough to determine the tag of a specific term. That means over fitting could occur and to control such kind of scenario over fitting likelihood is penalized with the term $-\sum_{i=1}^{n}\frac{\lambda_i^2}{2\sigma^2}$. The parameter δ^2 models the tradeoff between fitting exactly the observed feature frequencies and the squared norm of the weight vector. The smaller the values are, the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced. The notation of the likelihood function L(T) is reorganized as:

$$
\begin{aligned}
\mathcal{L}(\mathcal{T}) &= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1},y_j,\vec{x},j\right)\right)}{\sum_{\vec{y}'\in\mathcal{Y}}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1},y'_j,\vec{x},j\right)\right)} \right) \right] - \sum_{i=1}^{m}\frac{\lambda_i^2}{2\sigma^2} \\
&= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \left[\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1},y_j,\vec{x},j\right) \right) - \right. \\
&\qquad \left. - \log\left(\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1},y'_j,\vec{x},j\right)\right)\right)\right] - \sum_{i=1}^{m}\frac{\lambda_i^2}{2\sigma^2} \\
&= \underbrace{\sum_{(\vec{x},\vec{y})\in\mathcal{T}}\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1},y_j,\vec{x},j\right)}_{\mathcal{A}} - \\
&\qquad \underbrace{- \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \log \underbrace{\left(\sum_{\vec{y}'\in\mathcal{Y}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1},y'_j,\vec{x},j\right)\right)\right)}_{Z_{\vec{\lambda}}(\vec{x})}}_{\mathcal{B}} \underbrace{- \sum_{i=1}^{m}\frac{\lambda_i^2}{2\sigma^2}}_{\mathcal{C}} .
\end{aligned}
$$

4.

The partial derivations of L(T) by the weights for parts A,B and C are computed separately.

Derivation for part A:

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

5.

Derivation for part B, which corresponds to the normalization, is given by:

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

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

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

$$= \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}\in\mathcal{Y}} p_{\vec{\lambda}}(\vec{y}'|\vec{x}) \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j) \quad 6.$$

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

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \quad 7.$$

Equation 5, the derivation of part A, is the expected value under the empirical distribution of a feature f_i:

$$\frac{\partial}{\partial \lambda_k}\left(-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \quad 8.$$

Accordingly, equation 6, the derivation of part B, is the expectation under the model distribution:

$$E(f_i) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}'\in\mathcal{Y}} p_{\vec{\lambda}}(\vec{y}'|\vec{x}) \sum_{j=1}^{n} f_i(y'_{j-1}, y'_j, \vec{x}, j) \quad 9.$$

The partial derivations of L(T) can also be interpreted as

$$\frac{\partial \mathcal{L}(T)}{\partial \lambda_k} = \tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} \quad 10.$$

And the maximum likelihood method finds the parameter by equating the first derivation to 0 as:

$$\tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\delta^2} = 0 \quad 11.$$

Computing $\tilde{E}(f_i)$ is easily done by counting how often each feature occurs in the training data. Computing $E(f_i)$ directly is impractical because of the high number of possible label sequences. 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 complex to explain in this study [31].

2.6 Named entity recognition tools

There are tools which helps to label NEs, Natural language tool kit (NLTK), Stanford named entity recognition, GATE, Lingpipe… etc. NLTK is a python platform that is more suitable for Part of Speech (POS) tagging. Stanford NER is a Java implementation of a Named Entity Recognizer developed by Stanford University. When Stanford started NER their scope was focused on identifying Biological terms (genes and proteins) from biomedical papers [32]. After their first attempt, they have expanded their scope by increasing the classes and also with regard to features the model they have used changed from time to time till they implemented a sequence classifier to the unseen domains (i.e. CRF). We have used this customizable and open source NER tool for labeling our data.

Lingpipe is also 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 [33]. [6, 8] are used this tool.

2.7 Amharic language structure

In this section we have discussed language structure of Amharic with regard to our focused area, which is NE in Amharic. The section describes the background of Amharic script, word classes and characteristics of Amharic NEs. Even though, there are different word classes in Amharic we have only focused on Nouns and pronouns which are related to our work.

According to [5], 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

The present Amharic writing system was adopted from the Ge'ez writing system. Ge'ez, which belongs to the class of Semitic languages, was the language of literature in Ethiopia in earlier times [34]. The ancient Sabaean script is in turn attributed to the source of the Ge'ez script. However, as [34] explains, the number of symbols in the original Sabaean script and their shapes have changed into Ge'ez and then later on into Amharic. Moreover, some new symbols have been added to Amharic. The current Amharic writing are combination of all Ge`ez fidel and some added characters.

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. It consists of 33

consonants, giving 7*33=231 syllable patterns, or fidels. In addition to the 231 characters, there are other non-standard alphabets which contain special features usually representing labialization. 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. Refer to appendix-1 for a complete list of the symbols [35].

Word classes of Amharic language are classified in to different categories by different scholars. As Mersehazen(1935 e.c), cited on [36] put the word classes as eight. Those are preposition, noun, conjunction, interjection, verb, adjective, pronoun and verb. Whereas, the recent works (Baye, 1987 and 97) as cited by [36], reduced the previous classes in to five word classes. The reduction of Amharic word classes are based on different reasons, Baye's reduction of Mersehazen's classification seems based on the role of words in syntax, which means by considering the clear role of words in Amharic grammar.

2.7.1 The Noun Class

Like English, Amharic nouns are words used to name or identify any of a class of things, people, places, organization or ideas or a particular one of these [37]. Formations of nouns are either simple, augmented or compounds.
As [34] Discusses, the formation of nouns are described as follows:

Simple forms; consisting of two, three, or four letters.

A. Bilateral.
    a. Ending in the second order:
        ሙሉ፡ full.
        ጥሩ፡ good
        ክፉ፡ bad

        These all words are originating from or representing the passive participle.

b. Ending in the third order, generally signifying an agent:

መሪ ፡ guide

ሰፊ : wide

ሰሪ : workman

c. Ending in the fourth order:

ውኃ፡ water

ስጋ: flesh

ካራ: knife

d. Ending in the fifth order:

ቅቤ፡ butter, oil.

ጥሬ: genuine, original.

በሬ : ox.

e. Ending in the sixth order.

ቀን ፡ day

ላም : cow

ሙዝ፡ banana.

Most of bilateral nouns are ending in these order and they are numerous.

f. Ending in the seventh order:

ጉዞ ፡ a day's march.

ጎጆ ፡ small thatched house.

ቆሎ ፡ fried grain.

In triliteral and bilateral, [34] listed out different nouns. Noun words with more than two alphabets (fidel*)* are discussed in detail.[2]

[2] C. W. Isenberg, *Grammar of The Amharic Language*. London: Richard Watts 1976.

Compound nouns: are nouns which are written as two separate words and sometimes as a single word [38]. E.g. “ ቤተ-መንግስት” which mean palace. “ ትምህርት ቤት” school “ፅህፈት ቤት” and “ ፅፈትቤት “ which both mean secretariat office.

Derivative nouns- are nouns which repeat themselves in order to pluralize the quantity they are expressing for. E.g.

ፍሬ	ፍሬ-ኣ- ፍሬ	ፍራፍሬ
ቁስ	ቁስ-ኣ-ቁስ	ቁሳቁስ
ትል	ትል-ኣ-ትል	ትላትል

2.7.2 Pronouns

According to [34] pronouns are divided as in other languages, into:

1. Personal
2. Possessive
3. Demonstrative
4. Relative
5. Interrogative
6. Reflective and

7. Distributive Pronouns.

These classes are grouped into two categories, namely, Separable and Inseparable Pronouns.

Personal
Possessive
Demonstrative
Interrogative
} Separable

Relative
Distributive
} Inseparable

2.7.2.1 Separable Personal Pronouns

They are three for the Singular, and three for the plural. The singular has some peculiarities. The first Person has not the gender expressed, whereas, the second and third have distinct forms for the Masculine and for the Feminine gender. The second person has, besides, three distinctions of honor.

	Singular			Plural
	Masculine	common	Feminine	common
1 Person		እኔ: I		እኛ : we.
2 Person	አንተ: you		አንች: / አንቺ: (thou)	እናንተ: you.
		አንቱ / እርስዎ (you)		
3 Pers.	እርሱ: he,it.		እርስዋ: she, it.	እርሳቸው: they.

The above structure is Amharic separable personal pronoun formation and Detailed discussion on Amharic pronouns are found in [34].

2.7.3 Amharic Named Entities

According to [39], NEs are phrases that contain the names of persons, organizations and locations. In the expression named entity, the word named restricts the task to those entities for which one or many rigid designators stands for the referent.

Person names (PERSON)- name of a person including his/her father and grandfather

Location name (LOCATION) – name of geographical entities like cities, regions, country.

Organization name (ORGANIZATION) – name of entities like companies, institutions and organizations.

Amharic has no case information like English and other Latin character languages which ease identification of NEs. It is one of those properties used in the tool (Stanford Named Entity Recognition), but due to the previously mentioned property of Amharic we have excluded capitalization feature.

There are words that mostly co-occurred with NEs, Clue words are useful properties in identifying NE and that mostly precede NE words and sometimes they succeeded NEs. Clue words for each NE classes are different, some of them for PERSON are:

ፕሬዚዳንት	President
አቶ	Mr.
ወ/ሮ	Mrs.

ፕሬዚዳንት ግርማ ወ/ጊዮርጊስ	President Girma W/Giorgis
አቶ በዓሉ ግርማ	Mr. Bealu Girma
ወ/ሮ አዜብ	Mrs. Azeb

Organization NEs are most of the time succeeded with the following clue words.

ካፒታል ሆቴል	Capital Hotel
ሐረማያ ዩኒቨርሲቲ	Haramaya University
ፕሬዚዳንት ቢሮ	Office of president
የተባበሩት መንግስታት ድርጅት	United Nations Organization

Location NE classes have the same properties with Organization. In addition to clue words, Amharic NEs present with affixes.

Affixes are classified in to three, Prefix, Suffix and Infix.

Prefixes, suffixes and Infixes are sets of letters that are added to the beginning or end of another word. They are not words in their own right and cannot stand on their own in a sentence.

E.g. □□□□ □□□□□ □□□□□ □□□□□ □□□□

Esayas Afewerki is president of Eriteria.

□□□□□ □□□□□ □□□□□….

Esayas Afewrki's government…

On the first sentence /□□□□ □□□□□ / is a PERSON name, whereas, when we add the prefix /□/ the name changes it class to ORGANIZATION.

E.g. አዲስ አበባ ዩኒቨርሲቲ በኢትዮጵያ ዋና ከተማ ውስጥ ይገኛል፡፡

Addis Ababa University is found at the capital city of Ethiopia.

ከአዲስ አበባ ዩኒቨርሲቲ እስከ…

From Addis Ababa University….

Here in the first sentence the word /*አዲስ አበባ ዩኒቨርሲቲ* / is an ORGANIZATION name, but in the second form of the word /*ከአዲስ አበባ ዩኒቨርሲቲ* / becomes LOCATION name. The change comes after the addition of prefix /ከ/. N.B the second case is not always true, because the prefix by itself is not sufficient to identify NE classes. For eg.

አዲስ አበባ ዩኒቨርሲቲ በኢትዮጵያ ዋና ከተማ ውስጥ ይገኛል፡፡

Addis Ababa University is found at the capital city of Ethiopia.

ከአዲስ አበባ ዩኒቨርሲቲ የተማሪዎች መማክርት…

From the student council of Addis Ababa University…

In such cases the name/አዲስ አበባ ዩኒቨርሲቲ/ with prefix /ከ/ and the name /አዲስ አበባ ዩኒቨርሲቲ /in the first sentence have the same NE class, ORGANIZATION.

2.7.4 NER challenges for Amharic

Several studies indicated that NER is a difficult task and a number of challenges are still not addressed because of many reasons: Ambiguity, Agglutinative Nature of the Language, Spelling Variations and A Nested Named Entities**.**

2.7.4.1 Ambiguity

In Amharic there are proper names that have more than one NE class. ፀሐይ (*Sun)* can be person's name and it is used to name Sun. Amede, H/Giorgis,

Haile Gebresilase ,Abune Petros and Belay Zeleke are some of the names that are used for both location and person's name in Amharic. During model creation, handling ambiguity words are performed with a feature that can learn their contextual pattern in the sentence.

2.7.4.2 Agglutinative Nature of the Language

Agglutinative property means that some additional features can be attached to the word to add more complex meaning. According to [7],most of the time NE root words cannot be found in a document, rather they are found in their morphological derivations that are formed by affixes. As we have discussed in section 2.7.3 Amharic Named Entities prefixes helps to identify class of a word in Amharic. For example, when the prefix, ከ- adds to ኢትዮጵያ (Ethiopia) it becomes ከኢትዮጵያ (From Ethiopia) and we can used such kind of affix as a feature to classify that word under LOCATION class.

2.7.4.3 Spelling Variations

In Amharic there are words which can be written in different formats. For instance, "ፀሐይ" and "ጸሐይ", "ፀሀይ" and "ጸሀይ" all mean the same, "sun" although they are written differently (indicating alternate use of ፀ and ጸ, and ሀ and ሐ. In our research we have tried to handle such kind of language challenges by using transliterated data, in transliterated corpus all the afore mention names are written in a single format, *tShay* (Sun) and controlling spelling variations in NERs ease extracting features and applying it in all words that have found in the same format.

2.8 A nested NEs

It is a word that are formed by a combination of two names, but identifying NE class of a word without contextual consideration leads to wrong classification of NE class and less NER system performance. A word can be classified either of the three NE feature, but when it comes with some words it changes the previous NE class, such kind of problems are common in Amharic. For instance, አዲስ አበባ ዩኒቨርሲቲ Addis Ababa is location name, but when the word

ዩኒቨርሲቲ (University) ከተማ አስተዳድር (City Government) and some others are added it becomes Organization names. አዲስ አበባ ዩኒቨርሲቲ (Addis Ababa University) and አዲስ አበባ ከተማ አስተዳድር (City Administration of Addis Ababa). Whereas studying the co-occurred names during model creation overcome challenges that existed in nested NEs.

Chapter three

Related works

There have been different NER systems development attempts taken by different authors throughout the world. In this section an overview of NER will be discussed, by categorizing works on domestic and foreign language. In the domestic languages, one Amharic [6] and two Afan Oromo [8] [7] named entity recognition systems have been developed.

[6] Used CRF to develop a NER system in Amharic with internal and external features, Context features, Part Of Speech tags of tokens, prefix and suffix. The corpus contains 10405 tokens, was split into training, developing and testing data set. To conduct the experiments the researcher used tagged corpuses that are found from news articles, they followed supervised learning method. During the experiment identification of the most relevant features for Amharic NER were attempted and the reported accuracy at first was F-1 73.47% with all of features. When the system was tested by removing POS tag, Prefix and Suffix the performance was 69.70%, 74.61% and70.65%, respectively. The reason,[6] concluded, was that POS tag and Suffix are most relevant to identify Amharic NEs. Even though their research is conducted on an annotated corpus that are taken from WALTA news agency, POS tagging was trained by using Ling pipe tool it is a customizable environment developed for English.

[8] Used a supervised machine learning method and CRF model to develop a NER system in the Oromo language. The features used are both internal and external, normalized tokens, part-of-speech tags, word-shape features, position features, and token prefixes and suffixes. Adopted from CoNLL(Conference on Computational Natural Language Learning), four types of Oromo language NEs are identified: Person, Location, Organization and Miscellaneous. As they described due to compatibility issue with previously proposed work for Oromo

language POS tagging which is external feature in their work they have used ling pipe tool that is originally developed to English. The processing was classified in to pre-processing, training and recognition phase. They have reported that to all processing they have used news based Afan oromo corpus which consists of 23,000 tokens from "*Bariisaa*" and "*Kallacha Oromiyaa"* news papers. After generating the model they reported that on the first experiment the performance obtained are 77.41% Recall, 75.80% Precision and 76.60% F1-measure. They have conducted their experiments by adding the corpus size and reducing features. They found that recall and F1- measure increase while precision reduced while removing feature in the experiment has shown an inverse change in all performance measurements. Finally [8]reported that features play a vital role than the change in the training data size.

The other attempt in the Oromo language is the development of e Afaan Oromo Named Entity Recognition (AONER) system. As they reported, AONER used the same method and model that have been used by [8]. The corpus in [7] is also news articles that is collected from Oromia Radio and Television organization, it contains 30, 402 tokens. Features used to identify the tokens in to Organization, Person and Location formats are current word, N-grams, previous word, next word, maximum n-gram length, current word shape and surrounding word shape sequence features. The data is split in to training and test set to conduct the experiment. [7] conducted three experiments those are based on current word, the use of n-gram(left/right word) and word shape sequence to identify and detect the boundaries of named entities. As they reported, the overall performance achieved by using the above listed features is F1- measure of 84.71 on test data set. The identified features that have the greatest significance during the experiment are categorized regarding their indication of named entity type. For Named entity type they found that, contextual features (previous, next word, two word previous, and two words next to current word) are a good indicators, clue words are good in identifying all named entity types due to the reason that has the best predictive value for

classifying the NEs and finally as [8]concludes they have also summarized their work, combinations of context and word shape sequence are very important for Afaan Oromo NER.

[3]Used first-order Markov HM-SVM with linear kernel and Expectation Maximization(EM) on the Arabic NER with different feature sets, Tag of other occurrences (GLB), Previous Class (C-1), Trigger(TRG),list of unambiguous names(UNQ),list of most frequent non-person bigrams(BIAMB),effective preceding unigrams(UNIG), Most frequent non-person tokens(AMB), POS tagging (POS), Gazetteers (GAZ) and Lexical (LEX). The data used was ACE corpus is in the form of raw text files of news stories with stand-off annotation in XML files , from the corpus they annotated 70,000 words of the Arabic Tree Bank with their named entity classes. As they reported, Ten experiments have been taken place by interchanging the above listed features. they have tested the corpus with two algorithms they have selected, due to the use of different features in the two learning algorithms they did not want to compare the two algorithms but the result found to HM-SVM and EM are 74.49and 69.18, respectively. As [3]reported, best average performance was on the location class, as it was easier in terms of the context of POS tags and also due to the repetition of location entities in the corpus and worst result was on the organization class, since there was no gazetteer employed and also due to the syntactic structure of their naming systems that make them look like any other Arabic phrase. After repeated experiment the identified features those have a positive impact to all NEs are the class of other occurrences of the word being processed together with a list of triggers, class of the previous word and POS

[40] Used a hybrid (Gazetteers and Hidden Markove) model on Indian Languages. They identified four named entities; Person (PER), location (LOC), temple, River and rest are assigning other tag. During the experiment they have used limited sentences that are domain specific (Tourism) corpus. First their experiment tested the corpus by each model independently and scored 40.13% and 97.3% for Gazetteers and Hidden Markove model, respectively.

Hence [19] described that the performance is varied but the corpus size they have used for the two experiments is different, 100 (Gazzetteers) and 40 (Markove) sentences. Finally they concluded that when they applied the hybrid model it scored 98.375%, with this findings [19] reported, that this NER system with high accuracy is built, then this will give way to NER in all the Indian Languages and further an efficient language independent based approach can be used to perform NER on a single system for all the Indian Languages.

[2] Used supervised method with both annotated and unannotated data that was manually tagged at the University of Antwerp. sixteen systems are participated in CoNLL-2003 shared task , Maximum Entropy model is the most used model by a participants and Conditional Random Fields also used by one of the participants , the participants were used different models separately and with combination one with the other. Four types of named entities are tried to detect, persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. The CoNLL-2003 named entity data consists of eight files covering two languages: English and German. For each of the languages there is a training file, a development file, a test file and a large file with unannotated data The English data was taken from the Reuters Corpus and The portion of data that was used for Germen data, was extracted from the German newspaper Frankfurter Rundshau. Main features used by the sixteen systems that participated in the CoNLL-2003 shared task sorted by performance on the English test data. affix information (n-grams, bag of words, global case information, chunk tags, global document information, gazetteers, lexical features, orthographic information, orthographic patterns (like Aa0), part-of-speech tags, previously predicted NE tags, flag signing that the word is between quotes and trigger words. Finally they concluded that, the best performance for both languages has been obtained by a combined learning system that used Maximum Entropy Models,

transformation-based learning, Hidden Markov Models as well as robust risk minimization

Conclusion

Literatures that we have reviewed have shown that all of them have followed supervised learning method and statistical approaches, but they have used different features. It clearly shown in [8] and [7] that both of them used the same model and similar corpus with different features from which they have got different performance. From [2] we realized that applications of machine learning models interchangeably for the same or different language would not bring a significant change. Whereas an effective selection, identification and understanding of features and nature of languages are the most important points to come up with a high performance NER system.

CHAPTER FOUR

Design and implementation of the system

4.1 Design of ANER

In this section we have discussed the overall design of our system, Amharic Named Entity Recognizer (ANER). The first part of this section discusses about our data sets which is used for training and testing phases. The second section contains approach we have followed in the study. Then the general overview of the proposed system architecture from the perspective of the system's flow of operations in the form of processes is discussed. Finally, detailed explanation of phases along with subcomponents included in each phase and our system performance measurement are presented.

4.1.1 Data sets

As it has been defined in [6],[7]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. In this ANER, we have used transliterated 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. But we have chosen to use the transliterated one, as we have discussed in section 2.7.4.3, using transliterated corpus reduced spelling variations which are common in Amharic script (fidel) writing. 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. The corpus 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”

and this makes direct use of the corpus difficult. So, removing the string has taken place before going to the main preprocessing phase.

4.1.2 Corpus preparation for ANER

In our system we need two types of data, training and testing data. Training data are used to train the machine learning component. Then after, conducting training and creating a Model, the system has tested on testing data to evaluate the performance of the system. Before going to prepare our data we have been tried to get previously annotated data by [6]; both the researchers and the respective department were not able to provide us the needed data, after several attempt finally we gone to prepare our own data.

By using simple random sampling technique, 13538 tokens from WIC corpus were selected for both phases. During sample selection sentences with a minimum of one NE have been selected, this is with the assumption of if there is no NE in a training data, finding the pattern and generating a good model will be tough and testing such kind of model may have degrade the performance of a system.

After selecting the corpus and conducting preprocessing tasks, removing manually tagged POS and tagging with named entities based on Stanford notation were conducted. According to the format every word in a sentence will be kept on one line together with the NE tag separated by a tab space.

4.1.2.1 Data format

yemaIkelu O

astebabari O

ato O

belayneh PERSON

ayalEw PERSON

lewalta ORGANIZATION

InformExn ORGANIZATION

maIkel ORGANIZATION

IndegeleSut O

maIkelu O

" O

abobako O

" O

yemil O

syamE O

yeteseTewn O

yebeqolo O

zerna O

" O

1xi O

107 O

" O

yeteseNewn O

yemaxla O

zer O

yageNew O

balefew O

and O
amet O
bakahEdew O
mrmr O
new O
:: O

The data contains entity of three types: persons (PERSON), organizations (ORGANIZATION), and locations (LOCATION). Words tagged with O (Others) are words which do not belong to the above three named entities.

4.1.3 Approach Used

From the related works that we have reviewed [22], supervised learning is the current dominant technique for addressing the NER and ANER is designed by following this approach. The machine learning component learns from annotated training data. Conditional Random Fields (CRF) algorithm is the machine learning component we have used. A CRF algorithm has the efficiency of dealing with non-independent, diverse and overlapping features. It has also the capability of overcoming the label bias problem. These two factors make CRF algorithm better suited for NER. The preprocessed annotated data is given to the training phase to create a model, then, testing phase, which is the recognition of the named entities using the model that is generated by the training phase are performed. Figure 4.1 describes our architecture.

4.1.4 Architecture of ANER

Under this section, the architecture of the system will be presented. Figure 4.1 shows the overall functionality of the Amharic Named Entity Recognition system. There are three main phases in our ANER system, Preprocessing, training, and testing. These three major phases might be used in different NER and other machine learning systems, but tasks which are taken place within

each phase are different and this is caused by the data we used and the procedure that a researcher has chosen. Because of the above reasons, there is no single architecture that we all should follow. With this assumption our ANER architecture is described as follows.

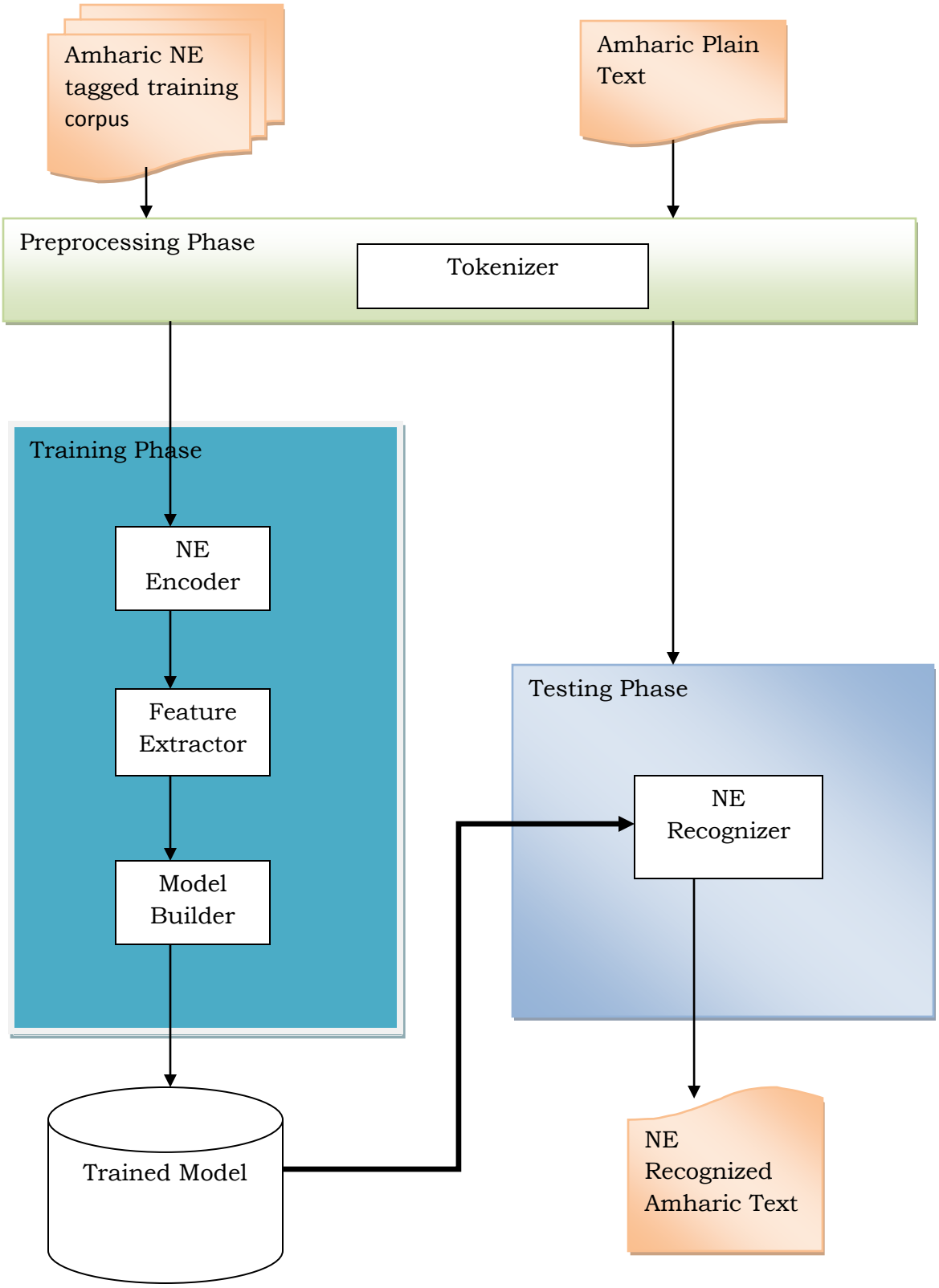

Figure 4.1 Architecture of ANER.

4.1.4.1 Processes of ANER

ANER is designed in a manner that, first it learns properties and parameters associated with NEs from the training data. It then receives input plain text and predicts the possible NEs out of the plain text. The architecture has two processes: learning process and prediction process.

4.1.4.2 Learning process

Training is handled by components in the learning process. Preprocessing is initially performed on the training corpus. The corpus is tagged based on Stanford NER format. Then it is passed to the NE encoder which identifies a token and its corresponding NE tag through encoding. The token and tag sequence generated by the NE encoder is handled to the Feature Extractor. Feature Extractor extracts necessary features to identify NEs based on the generated token and tag sequence, the extracted features will then be used as an input to the Model Builder. After the above all fulfilled, the builder begins the model building procedure to generate a trained model. Generally, here in the learning process, based on the training corpus we have generated a model that will be used to predict NE classes of testing.

Pseudo code for learning process:

For *all tokens in the training data*

Activate *previous word sequence feature*

Based on the specified maximum length of left and right window size

fExtractor .extract(previous word feature for the current token and set as 1)

Activate *next word sequence feature*

Based on the specified maximum length of left and right window size

fExtractor .extract(next word feature for the current token and set as 1)

Activate *word pair feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(word pair feature for the current token and set as 1).

Activate *word shape feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(word shape feature for the token and set as 1).

Activate *NE tag feature*

Based on the specified maximum length of left and right window size

fExtractor.extract(NE tag feature for the current token and set as 1).

fExtractor.append(the NE tag of the previous token and set as 1).

fExtractor.extract(the NE tag of the next token and set as 1).

Activate prefix and *Suffix feature*

Based on the specified maximum length of prefixes and suffixes

If *the length of the current token is less*

Ignore prefix& suffix feature for the current token.

Otherwise

fExtractor.extract (prefix & suffix feature for the current token and set as 1).

End For

Return fExtractor().

4.1.4.3 Prediction process

Here the plain text is preprocessed. Tokenization is the only preprocessing task performed on the input plain text. The tokenized data is then passed to the NE Recognizer. The trained model will assist the NE Recognizer to perform such tasks as calling stored features and NE tagging on the input data. The NE Recognizer recognizes NEs out of input text with the aid of the trained model. The result, NE recognized Amharic text is then supplied as an output.

4.1.4.4 Preprocessing

4.1.4.4.1 Tokenization

Tokenization can basically be defined as the process of breaking up a text into its part of elements, tokens. In Amharic, tokenization process is easier since every word is separated by a space. Tokenization is done on the input plain text during the prediction process. Tokenization on the training corpus is also done during corpus development. The tokenization takes the input text supplied from a user and tokenizes it into a sequence of tokens that can make it easy to recognize NEs.

4.1.4.5 Training phase

Training phase contains necessary components that are used in the process of training the machine learning component. The components within this phase operate on the data from the training corpus and they are the NE Encoder, Feature Extractor and the Model Builder.

4.1.4.5.1 NE encoder

This is a process that provides information to feature extractor by identifying a token and its tag. From the training data the encoder analyzes which one is a token and which one is a tag then supplies to the next process. Hence, the encoder provides important clue to feature extractor, setting it wrongly also leads to wrong model creation. To avoid this we have kept the consistency between corpus and NE encoder.

4.1.4.5.2 Feature extractor

As we have discussed in chapter two, Features are properties of a text that are used to provide necessary information associated to a given NE and increase the confidence level of predicting a token as a NE. ANER's feature extractor is responsible for identifying and extracting all the necessary features from the training data. It is one of the essential components that supply necessary information (features) during the building of the model. The feature extractor is designed to extract features from the training data and store them for future use.

The feature extractor that we have used, CRF algorithm, performs repeated feature extraction from the same context for different positions in the input. All the extracted features are supplied to the model builder which will predict the parameters of the model. To identify Amharic NEs we have used: Word Prefix and suffix, word shape features, word position features with a different window sizes and the detailed feature combination used are described in section 4.2. As we have discussed in chapter two POS tag feature is used in different NER systems and they have scored various performance, but with regard to the tool we have chosen (Stanford), the features used in our work are very similar to POS tags. Due to this reason we have focused on using other features that predict NEs of Amharic.

4.1.4.5.3 Model Builder

The main concern with the machine learning component is to build a trained model that will be used in prediction. Building a model is the component designed to estimate the model Coefficients and then build a trained model. Estimation is taken place based on the input from the training corpus that is supplied by extracted features which is generated from feature extractor. The model builder is designed to perform all the calculations we have discussed in section 2.5 and generated a trained model that will be used in prediction phase.

4.1.4.6 Prediction phase

This prediction or recognition is the final phase in NER system. It is a process of recognizing NEs from the given preprocessed token. There are two sub tasks are taken place in this phase, detection and classification.

4.1.4.6.1 Detection

Detection is the first task which finds candidate tokens that can be NEs from the tokenized plain text. By using the knowledge acquired from the model, detection is performed. The knowledge in model builder contain features that are extracted and stored during the training phase, are supplied to the recognizer to identify NEs from the given plain text. Identification of NEs are performed based on the calculated probability.

4.1.4.6 .2 Classification

After detection of NEs for the candidate tokens are performed, classification process is preceded by tagging the detected candidate tokens. During classification each token in the given plain text will be tagged with their possible NE tags using Stanford scheme. Finally, the recognizer will generate NE recognized Amharic text and send it as an output.

4.1.5 Performance evaluation

As we have discussed under Corpus preparation for ANER, for both phases (training and testing) 13,538 tokens were used. For training and testing we have used, 12296 and 1242, respectively. 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).

$$P=\frac{T_P}{F_P+T_P}$$

And

$$R = \frac{T_P}{F_N + T_P}$$

Where, True Positives (T_P) are NE tags which are produced by the model that match the reference tagging , False Positives(F_P) are NE tags returned by the model that are not in the reference tagging.
And lastly, False Negatives (F_N) are NE tags in the reference tagging bur they are missed by the model.

The F measure which is computed by the uniformly weighted harmonic mean of precision and recall:

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

Performances that we have discussed in the following section are performed according to these performance measurements.

4.2 Experimentation Result

In this section we have discussed experiments which are conducted in different scenarios. The different scenarios have taken by varying our feature sets. The first section presents corpus used in the experiment then describing feature set in the experiment. In the second section experimentations with their performance achievement are presented. The final section discussed experiment results with their graphical representation.

4.2.1 Corpus size

The number of tokens used in this work are 13538, from these 10% of the total tokens are used for testing and the rest are supplied for model creating.

Table 4.1 corpus size

Number of tokens in training phase	Number of tokens in testing phase	Total number of tokens
12296	1242	13538

Feature sets plays a crucial role in CRF frameworks, and as we have stated under objective of the study, finding the optimal feature sets for Amharic language is our intended objective. To find these feature sets we have used word based features contains suffixes and prefixes of the current and surrounding tokens and context based features that are previous and next words of a particular word with their named entity tags.

- Suffixes- 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 four of the current and surrounding word have been taken for this language since it gave better performance than using length up to 2.

- Prefixes - A fixed length prefix of the current and the surrounding words might be treated as features, in this work, the tool we have used did not allow us to use prefix and suffixes separately.
- Previous word- words that occurred before to a current word and it might be also a clue word that indicates NE class of succeeding word.
- Next word- words that occurred next to a current word and it might be also a clue word that indicates NE class of preceding word.
- Named Entity Tags- the NE tags are used in our work as a feature, these features are used by increasing our window size up to 4.
- Orthographic – word shape of a token was one of our features set. The experiment also conducted by increasing the window size for orthographic feature and combining it with other features.
- Word pairs - features that are taken by calculating frequency co-occurrence of words.

The above listed features are used in this work, but before deciding feature set of this system determining the first feature set for our baseline experiment was the toughest task during experimentation. This raised due to a reason that the earlier ANER system feature set are not compatible with the tool we have chosen, Lingpipe which is a tool that has been used by previous ANER researchers. It accepts suffix and prefix independently, beginning of sentence (BOS) and End of a sentence (EOS) are also additional features in their tool. The other feature is Part Of Speech tagging (POS), the feature that we have not used. As we have discussed previously, Stanford features have a capability that POS can performs. According to Stanford NER developers group, the features used by POS tagger are very similar to those used in the NER system, so there is very little benefit to using POS tags[3].

With all the above mentioned factors, using previous work features as a baseline experiment were impossible. Hence, starting our experiment with all

[3] http://www-nlp.stanford.edu/software/crf-faq.shtml#pos

the same features that have been used in the previous work is difficult we have selected common features from them. But feature usages are still different.

For our baseline experiment we have selected the following features that have been used and brought the optimal performance in previous work.

Table 4.2 Baseline experiment

Experiments	Features used	Performance		
		Recall %	Precision%	F-measure %
Baseline	Window size (left and right)=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag	76.84	82.95	79.78

NB. The features that we removed from previous work are BOS, EOS and POS, performance achieved by features in the table was F-measure of 74.61 % and it was the highest performance in their work. This performance was achieved with 10405 total token sizes.

After we have selected baseline experiment features for our work, identifying optimal feature set for ANER is continued. In order to begin the first experiment, we have started by selecting a single feature with a single window size; this is because of the assumption that studying a role of each feature independently would help to come up with optimal feature sets. With this reason previous word feature with one window size has chosen for the first experiment.

Table 4.3 Experiment one

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
1.	Left Window size=1 Previous word	52.6	75.94	62.15

The second experiment is conducted by increasing the window size of previous experiment by one that is two. Window size increments added was on the left side only and F-measure increased by 0.20%.

Table 4.4 Experiment two

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
2	Left Window size=2 Previous word	52.6	76.51	62.35

From table 4.4 we understood that window size increment has performed better than a single window size so, the next experiment used bigger window size. The third experiment was conducted by taking next word along with the above window size. The F- measure is the same as experiment 2 (i.e. previous word with window size two).

Table 4.5 Experiment three

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
3	Left Window size=2 Next word	52.6	76.51	62.35

After analyzing the influence of next and previous word feature independently the next step was studying the role of these two features combination. Combination of two features is taken with a window size of two to the left side.

Table 4.6 Experiment four

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
4	Left window size=2 Previous word Next word	52.33	75.94	61.97

As it is shown in Table 4.6 F- measure is reduced by 0.38% from F-measures achieved when features applied independently. Even though the F- measure is reduced, using previous and next word in combination has its own advantage. F-measure was not the only decreasing performance recorded but also Recall and precision too. Based on experiment 4 additional features are selected. Now, the identified features which positively recognized NEs are two window sizes to the left, previous and next words. The next experiment was conducted by using suffix and prefix with a length of two and four. As we have discussed in section 2.7.4.3**,** this work used transliterated news and most Amharic scripts are represented by two English alphabets. With this assumptions our instant prefix and suffix length started from two and finally reached to four.

Table 4.7 Experiment 5 and 6

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
5	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 2	63.87	75.31	69.12

6	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4	65.46	76.05	70.36

In table 4.7 the F- measures of our system has increased on both experiments, experiment 5 and 6 by 6.77 and 8.21, respectively. Then, based on the above result we have preceded to the next experiment by adding previous and next NE tags independently and combination of them to experiment 6. From our previous experiment (i.e. experiment 6) results in experiment 7, 8 and 9 shows, F- measure has increased by 9.42, 7.69 and 9.97, respectively.

Table 4.8 Experiment 7,8 and 9

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
7	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag	76.84	82.95	79.78
8	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Next word tag	76.19	80	78.05
9	Left window size=2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag	75.52	85.8	80.33

Here in experiment 9, the F- measure the system scored increases the baseline experiment by 0.55%. The next experiment has taken all the features in experiment 9 with a new window size. In experiment 10 we took window sizes of two, 1 to the left and right side of the current word.

Table 4.9 Experiment 10

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
10	Window size (left and right)=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 2 Previous word tag Next word tag	76.84	84.88	80.66

The F-measure of experiment 10 is increased by 0.33 % from our previous experiment result. The following experiment is conducted to show the role of other additional features. First we took all features from experiment 9 then, add orthographical and word pair features independently, finally combination of these two features has also been tested as follows:

Table 4.10 Experiment 11, 12 and 13

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %
11	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape	75.52	85.8	80.33

12	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word Pairs	74.48	86.67	80.12
13	Left Window size= 2 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape Word pairs	74.1	86.67	79.9

The above three experiments have shown that , orthographical features brought the same F- measure , Precision and Recall as features which have been used in experiment 9. Whereas , experiment 12 which is addition of word pairs feature only and experiment 13 , combination of word pair and orthographical feature have scored less F- measure performances than the previous experiment. During experiment 12 and 13 the F- measure recorded was reduced by 0.21% and 0.43 %, respectively.

The final experiment that we have conducted has taken all features in experiment 13 with one additional window size to the left. Here in experiment 14 F- measure has increased by 0.1% from experiment 13.

Table 4.11 Experiment 14

Experiment	Features used	Performance		
		Recall %	Precision %	F1 %

14	Left Window size= 2 Right window size=1 Previous word Next word Prefix Suffix Maximum length (suffix and prefix)= 4 Previous word tag Next word tag Word shape Word pair	76.84	83.43	80

4.2.2 Discussion of Experiment results

The above experiments show that different combination of features set have brought various F- measures, Recall and Precision. The following table presents performance of the systems.

Table 4.12 Performance

Experiments	Performance		
	Recall %	Precision%	F-measure %
Baseline	76.84	82.95	79.78
1	52.6	75.94	62.15
2	52.6	76.51	62.35
3	52.6	76.51	62.35
4	52.33	75.94	61.97
5	63.87	75.31	69.12
6	65.46	76.05	70.36
7	76.84	82.95	79.78
8	76.19	80	78.05
9	75.52	85.8	80.33
10	76.84	84.88	**80.66**
11	75.52	85.8	80.33
12	74.48	86.67	80.12
13	74.1	86.67	79.9
14	76.84	83.43	80

From baseline to the last experiment that is experiment 14, removing and increasing the number of features sets and window size have been taken placed. Removing and adding of features are followed researchers best rationality.

Features we have used in the baseline experiment are selected by the method that we have mentioned above.

To select the first experiment features we started from the baseline experiment and study the role of each feature in ANER. With this assumption analyzing the result of previous and next word features are conducted by changing the window sizes.

By using of previous and next words independently with the same window size we have achieved the same precision, recall and F- measure. After studying the performance from experiment 1, 2 and 3, the features identified for the next level were previous word, next word and window size of two from the left side of a current word. The main finding here is, a different window size have changed the performance in each steps and this clearly shows previous and next word features independently with a window size of two have performed better than a single window size.

Based on this understanding experiment 4 is conducted and it achieved less performance. This experiment shows, a combination of previous and next word with different window sizes degrades performance of ANER system. After conducting different combination of the above three feature sets, looking for the next feature set is continued by using all features that have been used in experiment 4.

In experiment 5 and 6, the length of prefix and suffix we took shows a great change from previous experiments. By varying lengths of these two features are also indicated there is a change in the performance. From these two experiments we have observed that prefix and suffix could be one of the optimal features in ANER system. Then, by using features in experiment 6 we continued for further optimal features and studying the role of previous and next word tags of a current token were taken placed.

Here in the following experiments we have followed the same procedure as previously conducted experiments. From the three experiments that take each feature independently and their combination shows a tremendous change from experiment 6. As we have observed in the experiment, the combination of previous and next tag performs better than performances that were recorded when they are added independently. But previous tag performs better than next word tag.

After experiment 9, we have gone back to the baseline experiment, this is because of in experiment 9 combination of previous and next tag outperforms on applying previous tag only. By taking the result of experiment 9 into consideration, experiment 10 was conducted by adding next tag to features in the baseline experiment and we got the highest performance of the entire experiments.

Finally , based on the performance achieved in experiment 9 finding other optimal features are supported by adding orthographical feature, word pair feature and combination of them with different window size were conducted. Performance achieved in four of the above experiments shows degradation from the previous one and the researchers reached to, a simple addition of many features could not improved the performance of ANER system.

The general graphical representation of the above experimentation discussion is presented by clustering our fifteen experiments in to five clusters. Clustering is based on feature set which have been used in the experiments. The first cluster contains experiment 1, 2, 3 and 4; those are conducted by using previous and next word features with different window size. Experiment 5 and 6 are grouped in to second scenario that clearly shows performance changes over prefix and suffix features with different length. Previous word tag and next word tag used in experiment 7, 8 and 9 are clustered in to the third scenario. The fourth cluster (experiment 11, 12, 13 and 14), holds word pair and word

shape feature sets with different window size and finally baseline experiment and our highest performance experiment (i.e. 10) are fallen in to cluster 5.

Based on the above clustering , first the change within each clusters are presented then the second graphical representation held five experiment results by taking transition performance, which is an experiment performance that we added a feature to precede for the next experiment. In the second type of graphical representation the researchers took experiment 14 from the fourth cluster, this is because there was no transition after this experiment and this is one of the experiment that used all feature set of ANER system.

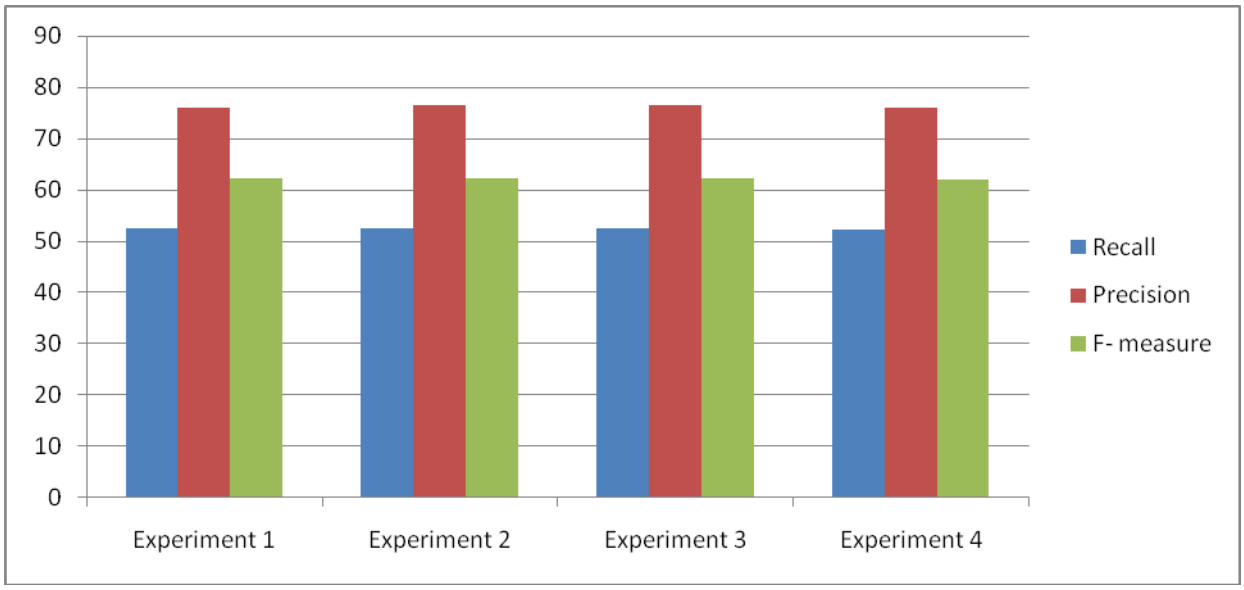

Figure 4.2 Scenario 1.

The first scenario shows F- measure did not increase when we change window sizes of previous and next word features .

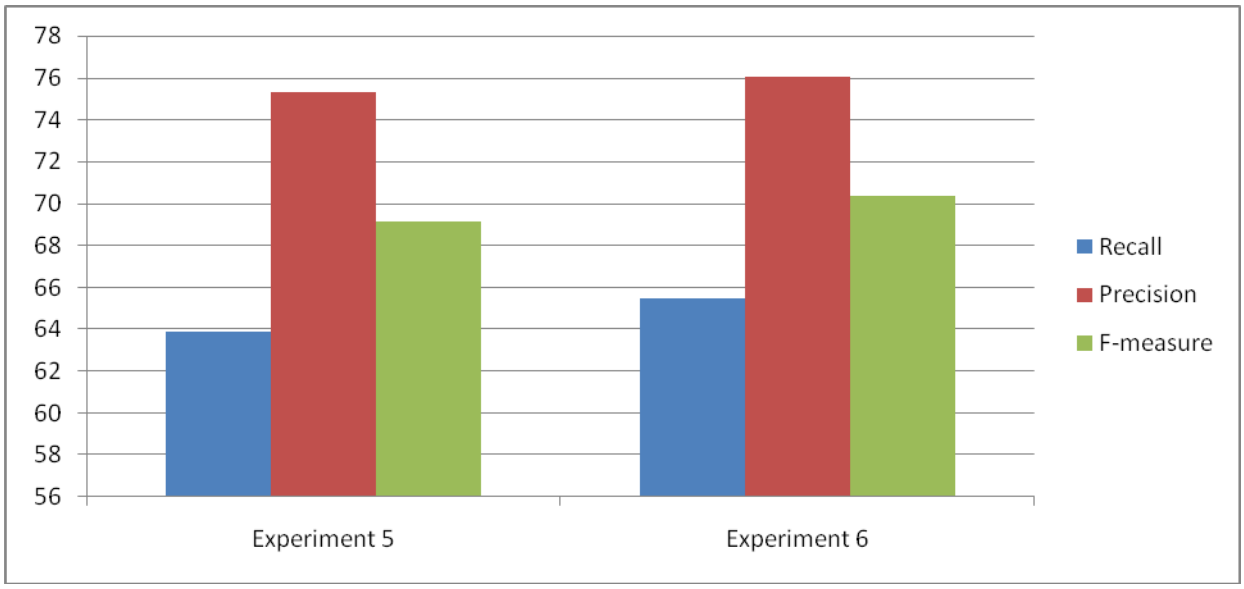

Figure 4.3 Scenario 2

It is our first obsrvation that , prefix and suffix have plyed a role in identifying Amharic NEs. as it is shown in the aove graph within each experiments suffix and prefix length also alter NE recognizing ability of a system.

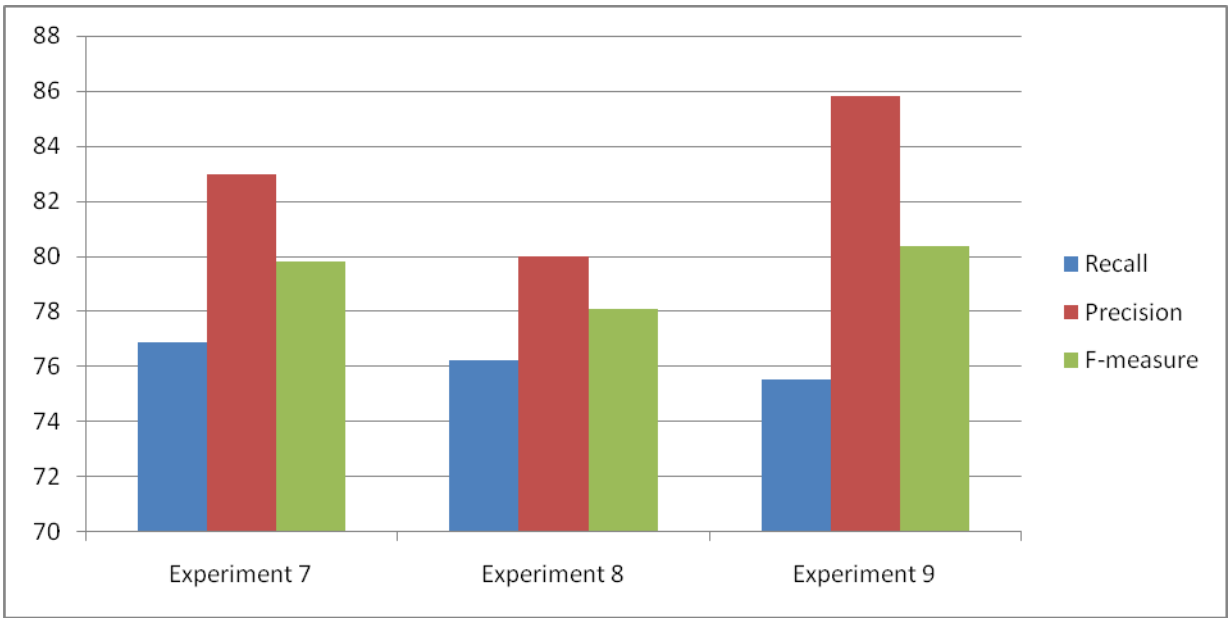

Figure 4.4 Scenario 3

After prefix and suffix , the drastic change has been obsrved during previous and next tag features. the change in each feature has changed . The worst perfrmance during scenario 3 is independent usage of next word feature and the maximum was combination of the two features. the precision in these scenario is one of the greatest from previous experiments.

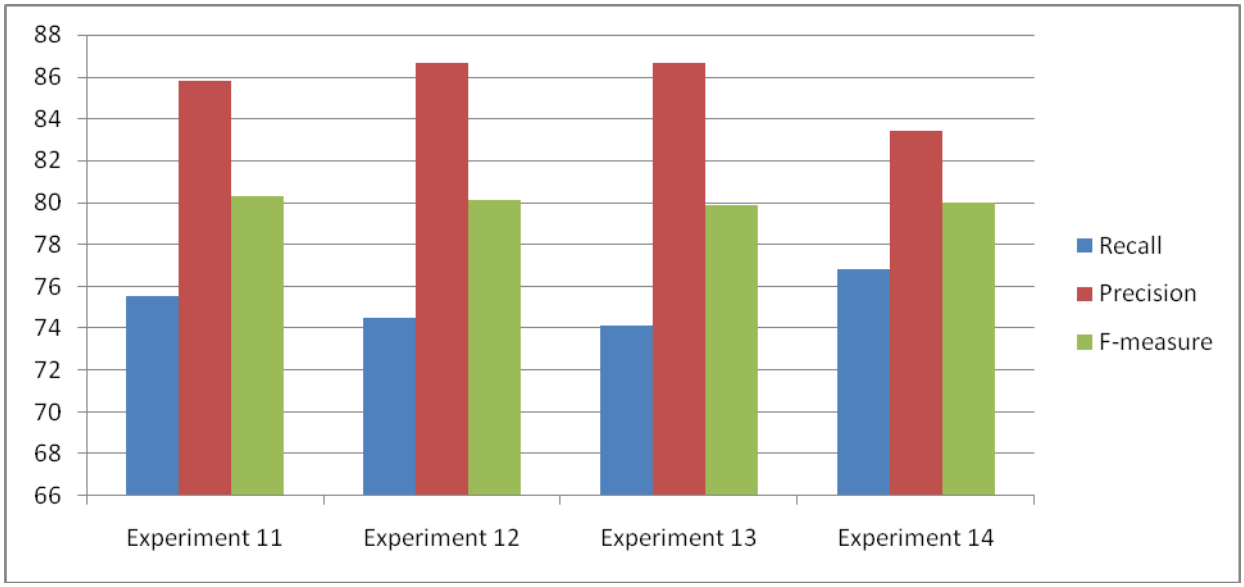

Figure 4.5 Scenario 4

As compared to scenarion3 the change in scenario 4 is insignificant . Recall and precision in scenario 4 are varied but F- measure in experiment 11 repeats experiment 9 result, where as the rest experiments performed with less F- measure that experiment 11 scored. The performance in scenario 4 indicated after certain incriment of feature set the performace will saturate and with worst case it inclined to reducing F- measure of the system . the reason behind this performance reduction need further study.

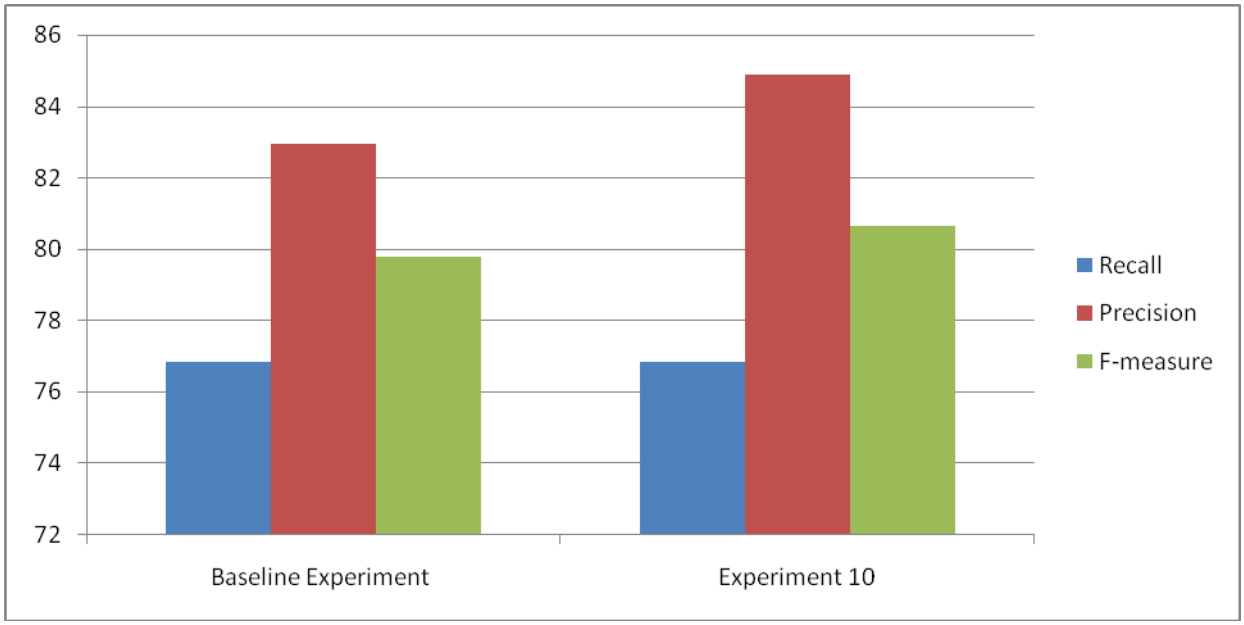

Figure 4.6 Scenario 5

The final cluster in the first graphical representation is scenario 5. In this scenario precision, recall and F- measure changes from baseline to experimt 10, as it is mentioned in scenario 3 combinational usage of previous and next tags have a potential to recognize Amharic NEs and the new finding in this scenario is , performace of a system varied in combinational usage of previous and next tags with different window sizes.

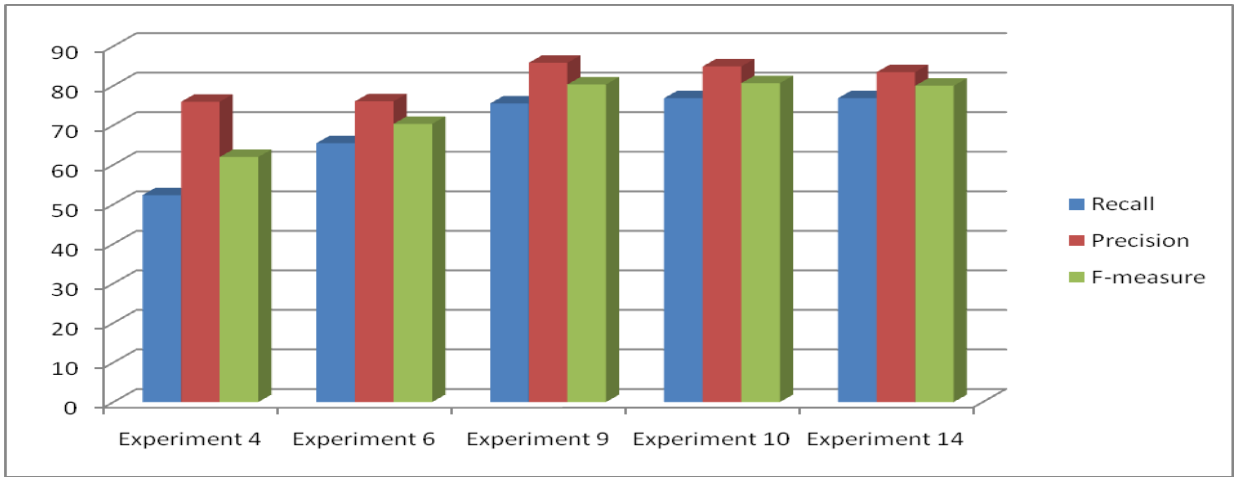

Figure 4.7 Second scenario graphical representation

From the above graph we understood that, Recall, Precision and F- measure of the system have been changing from experiment 4 to 6 and 9 in a tremendous way then it increased with a little decimal value in experiment 10. Finally the performance of the system reduced in experiment 14.

Generally, during our experimentation the identified optimal features that bring tremendous change are : suffix and prefix with 4 word length , previous and next word tag of a current word and window size of features.

Chapter five

Conclusion and recommendation

5.1 Conclusion

Named entity recognition is the process of identifying the right tag for names in a document. Many NER systems are developed for English and other European languages, next to these, NER for Arabic and south Asian languages have been developing. Amharic is one of the languages that have many speakers in Ethiopia, but it is under resourced.

This work tried to contribute one important point which plays a role for overcoming some challenges in NLP regarding to Amharic. Having a good NER system could be used in information extraction of any domain specific tasks, question answering, information retrieval … etc. to come up with a system that could be used in the above application areas, identifying optimal feature set is the most important task of any NER. This work is also delimited to identifying these features to recognize PERSON, ORGANIZATION and LOCATION names of transliterated Amharic corpus.

This study has proposed and designed a system called Amharic Named Entity Recognition. The system is designed based on a supervised machine learning component. We implemented the machine learning component using CRFs algorithm. To do this research, 13538 tokens were taken from WIC corpus and tagged manually. These data sets are classified into training and testing. By using the data processes of identifying optimal features for ANER has have been implemented, we have conducted fifteen experiments and the roles of each feature in different experiments were tested. The highest performance achieved in this work is 80.66%, with a window size of two on both sides, previous and next tag of a current word and prefix and suffix with length four. The worst performance achieved is 61.97%, feature sets are a window size of

two from the left side of a word and previous and next word of the current token.

The previous work on ANER has performed with 74.61 on a different tool, corpus size and feature set. Due to these reasons we could not compare this work with the previous one. But from the finding we got, a tool might have impact on the performance of ANERs.

From the findings, prefix and suffix with a length of four, previous and next NE tag of a token and window size features are the observed optimal feature sets for recognizing Amharic named entities.

During combining a feature set one needs to be careful in grouping the features; otherwise it degrades system's predicting ability.

5.2 Recommendation

This study showed some of the features that have never got an attention of previous ANER. The involvement of those new features has brought a promising performance and the following points are some of our observation that should be taken in consideration for future works of ANER.

- Due to the limitation of the tool that is used for this study, we could not observe the separate role of prefix and suffix. So, separating these two features and identifying whether their combinations or independent role should be included in ANER optimal feature set is recommended.
- Experimentation with regard to corpus size has never been conducted in this work and the corpus size in this work is very small with regard to CRF based NER systems and training by different corpus size would give a better performance
- From literatures we have reviewed[7, 8] used Lingpipe and Stanford, respectively and performance achieved with different corpus size and

features sets were 76.6 and 84.71, respectively , these shows further research regarding to NER tool needs to be.

- Developing a hybrid approach which incorporates a rule based on the nature and pattern of Amharic NEs with CRF based approach is recommended since it might give a better performance.

References

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes,* vol. 30, pp. 3-26, 2007.

[2] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 142-147.

[3] S. M. Algahtani, "Arabic Named Entity Recognition: A Corpus-Based Study," 2012.

[4] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1-7.

[5] A. A. Argaw and L. Asker, "An Amharic stemmer: Reducing words to their citation forms," in *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, 2007, pp. 104-110.

[6] M. A. Mehamed, "Named Entity Recognition For Amharic Language," Msc., Computer Science, Addis Ababa University, Addis Ababa, 2010.

[7] H. Keno, "DESIGNING A NAMED ENTITY RECOGNITION SYSTEM FOR AFAAN OROMO TEXT," MSC, Information Science, Addis Ababa University, Addis Ababa, 2012.

[8] M. L. Kejela, "Named Entity Recognition for Afan Oromo," Computer Science, Addis Ababa University, Addis Ababa, 2010.

[9] S. S. Keerthi and S. Sundararajan, "CRF versus SVM-struct for sequence labeling," Technical report, Yahoo Research2007.

[10] D. Li*, et al.*, "Conditional random fields and support vector machines for disorder named entity recognition in clinical texts," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, 2008, pp. 94-95.

[11] P. Thompson and C. Dozier, "Name searching and information retrieval," in *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, 1997, pp. 134-140.

[12] T. Sehgal*, et al.*, "NLP & its Applications."

[13] A. Mikheev*, et al.*, "Named entity recognition without gazetteers," in *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 1999, pp. 1-8.

[14] E. Riloff and J. Lorenzen, "Extraction-based text categorization: Generating domain-specific role relationships automatically," *Natural language information retrieval,* pp. 167-196, 1999.

[15] J. Cowie and W. Lehnert, "Information extraction," *Communications of the ACM,* vol. 39, pp. 80-91, 1996.

[16] M.-F. Moens, *Information extraction: algorithms and prospects in a retrieval context* vol. 21: Springer, 2006.

[17] K. Kaur and V. Gupta, "Name Entity Recognition for Punjabi Language," *Machine translation,* vol. 2, 2012.

[18] R. Farkas*, et al.*, "The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning---Shared Task*, 2010, pp. 1-12.

[19] A. Kao and S. R. Poteet, *Natural language processing and text mining*: Springer, 2007.

[20] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning,* vol. 3, pp. 1-130, 2009.

[21] O. Chapelle*, et al.*, *Semi-supervised learning* vol. 2: MIT press Cambridge, 2006.

[22] D. Nadeau, "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision," Phd, University of Owtawa, Canada, 2007.
[23] E. Alpaydin, *Introduction to machine learning*: The MIT Press, 2004.
[24] J. Mayfield*, et al.*, "Named entity recognition using hundreds of thousands of features," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 184-187.
[25] M. Tkachenko*, et al.*, "Named entity recognition: Exploring features," in *Proceedings of KONVENS*, 2012, pp. 118-127.
[26] Y. Benajiba*, et al.*, "Arabic named entity recognition using optimized feature sets," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 284-293.
[27] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 2009, pp. 147-155.
[28] A. Ekbal and S. Bandyopadhyay, "Bengali named entity recognition using support vector machine," in *Proceedings of workshop on NER for south and south east Asian languages, 3rd international joint conference on natural languge processing (IJCNLP)*, 2008, pp. 51-58.
[29] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 2003, pp. 134-141.
[30] R. Klinger and K. Tomanek, *Classical probabilistic models and conditional random fields*: TU, Algorithm Engineering, 2007.
[31] J. Lafferty*, et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.
[32] B. Sun, "Named entity recognition: Evaluation of Existing Systems," Norwegian University of Science and Technology, 2010.
[33] M. Marrero, "Advances in Computational Linguistics," *Research in Computing Science,* vol. 41,2009, pp. 47-58, 2009.
[34] C. W. Isenberg, *Grammar of The Amharic Language*. London: Richard Watts 1976.
[35] B. M. HAILEMARIAM, "N-gram-Based Automatic Indexing for Amharic Text," Msc, SCHOOL OF INFORMATION STUDIES FOR AFRICA, Addis Ababa University, Addis Ababa, 2002.
[36] G. A. Demeke and M. Getachew, "Manual annotation of Amharic news items with part-of-speech tags and its challenges," *Ethiopian Languages Research Center Working Papers,* vol. 2, pp. 1-16, 2006.
[37] A. A. Argaw, "AUTOMATIC SENTENCE PARSING FOR AMHARIC TEXT AN EXPERIMENT USING PROBABILISTIC CONTEXT FREE GRAMMARS," Msc, Computer Science, Addis Ababa University, Addis Ababa, 2002.
[38] L. Bender and C. Ferguson, "The Ethiopian writing system," *Language in Ethiopia. Edited by ML Bender, JD Bowen, RL Cooper, and CA Ferguson. London: Oxford University press,* 1976.
[39] T. Zhang and D. Johnson, "A robust risk minimization based named entity recognition system," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 204-207.
[40] N. Jahan*, et al.*, "Named Entity Recognition in Indian Languages Using Gazetteer Method and Hidden Markov Model: A Hybrid Approach."