ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

AFAAN OROMO NAMED ENTITY RECOGNITION USING HYBRID APPROACH

Abdi Sani Genemo

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

March 2015

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

AFAAN OROMO NAMED ENTITY RECOGNITION USING HYBRID APPROACH

Abdi Sani Genemo

Signature of the Board of Examiners for Approval

Name	**Signature**
1. Dida Midekso (PhD), Advisor	_____
2. Yaregal Assabie (PhD)	_____
3. Solomon Atnafu (PhD)	_____

Acknowledgements

I would like to sincerely thank all those people who have made this thesis possible. First of all, special thanks are due to my advisor **Dr. Dida Midekso**, who monitored my work and took effort in reading and providing me with valuable comments starting from title selection to this end. Thank you very much!

Besides, I would like to thank**Alemgena B., Gobena H., Abdulqadir A. and Dejene G.** from linguistic department for their willingness to give me comments at all times without complaint on linguistic aspects and for providing me linguistic materials and in evaluating the system.

I equally wish to extend my appreciation and thanks to my friends **Birhanu B., Mamo M., Tegenu Y., Gemechis A., Gemechis T., Gelane F., Aziza Mohammed** and **Abdisa Haji** for their moral support and encouragement. And last, but absolutely not least, I thank God for giving me the wisdom and the strength I need to discharge my duty.

Table of Contents

List of Figures

List of Tables

List of Acronyms & Abbreviations

ACE	Automatic Content Extraction
ANNIE	A Nearly New Information Extraction System
AONER	Afaan Oromo Named Entity Recognition
API	Application Programming Interface
BIO	Begin Inside Outside
BOS	Beginning Of Sentence
CoNLL	Conferences on Natural Language Learning
CREOLE	Collectionof REusable Objects for Language Engineering
CRF	Conditional Random Fields
FNs	False Negatives
FPs	False Positives
GATE	General Architecture for Text Engineering
HMM	Hidden Markov Model
IE	Information Extraction
IR	Information Retrieval
IDE	Integrated Development Environment
JAPE	Java Annotation Pattern Engine
LHS	Left- Hand - Side
MEMM	Maximum Entropy Markov Model
ML	Machine Learning
MUC	Message Understanding Conference
NEs	Named Entities
NER	Named Entity Recognition
NERC	Named Entity Recognition and Classification
NL	Natural Language
NLP	Natural Language Processing
POS	Part Of Speech
POST	POS Tagger
RHS	Right- Hand- Side
SVM	Support Vector Machine

Abstract

Named Entity Recognition and Classification (NERC) is an essential and challenging task in Natural Language Processing (NLP), particularly for resource scarce language like Afaan Oromo(AO). It seeks to classify words which represent names in text into predefined categories like person name, location, organization, date, time etc.Thus, this paper deals with some attempts in this direction. Mostly researcher have applied Machine Learning for Afaan Oromo Named Entity Recognition(AONER) while no researchers have used hand crafted rules and hybrid approach for Named Entity Recognition(NER) task.

This thesis work deals with AONER System using hybrid approach, which contains machine learning(ML) and rule based components. The rule based component has parsing, filtering, grammar rules, whitelist gazetteers, blacklist gazetteers and exact matching components. The ML component has ML model and classifier components. We used General Architecture for Text Engineering (GATE) developer tool for rule based component and Weka in ML part. By using algorithms and rules we developed, we have identified Named Entity (NE) from Afaan Oromo texts, like name of persons, organizations, location, miscellaneous.Feature selection and rules are important factor in recognition of Afaan Oromo Name Entity (AONE). Various rules have been developed like prefix rule, suffix rule, clue word rule, context rule, first name and last name rule.

We have used AONER corpus of size 27588, which is developed by Mandefro [1].From this corpus we have used corpus of size 23000 for training and 4588 for testing of our work. And we havean average result of 84.12% Precision, 81.21% Recall and 82.52% F-Score.

***Keywords**: Named Entity Recognition, Named Entities, GATE Developer, Weka, Afaan Oromo.*

Chapter One: Introduction

1.1. Background

Language is a central means of communication that enables human beings to exchange ideas and information in their day to day activities. It is the only gift that identifies human beings from the rest of life [2]. It serves as a medium for negotiating and recording of cultural knowledge and ideas. In written form, language helps to preserve and convey knowledge over time and space, and it serves as a means of coordinating our day-to-day life with others in its spoken form [1, 2].

NL is a product of the human mind. Communications by human beings are known as natural language [3]. It is part of our overall cognitive make-up and relates to perception, reasoning, imagination, and, most importantly, to the experience of our body and the world around us.Furthermore, natural language is the embodiment of human cognition and human intelligence. It is very evident that natural language includes an abundance of vague and indefinite phrases andstatements that correspond to imprecision in the underlying cognitive concepts [27]. In computing, natural language refers to a human language such as English, Russian, German, or Japanese as distinct from the typically artificial command or programming language with which one usually talks to a computer. The term refers to both written and spoken languages[17].

NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications [1, 29].NLP is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech as it is spoken. NLP researchersdeal with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using human languages instead of computer languages.The applications of NLP include a number of fields of studies, such as machine translation, natural language text processing and summarization, user interfaces, information retrieval, speech recognition, artificial intelligence and expert systems, and so on.

The researchers of NLP use natural language (NL)technologies that convert human languages into formal semantic representations which computer applications can interpret, act on, and respond with easily understood grammatical sentences.GATE and LingPipe are some of the NL technologies used by the NLP researchers. Combining the scalability and processing power of computers with natural language understanding leads to many highly useful applications such as NER and automated question answering. As a result,people can interact more naturally with computer-based information using normal, familiar expressions instead of using carefully constructed computer jargon.

As the amount of information grows, the problem of managing the information becomes challenging. To solve this problem, Information Extraction Technology,which automatically transforms unstructured textual data into structured representation that can be interpreted and manipulated by machines, is needed.Example: NER technology, which enhance and enrich the use of information retrieval systems such as search engines and helps as pre-processing task for many other information extraction tasks such as machine translation, information retrieval, and question answering[25].

NE is a term or phrase that identifies an object from a set of other objects with similar attributes. Examples of named entities are names ofperson, companies, geographic locations, dates, address, phone numbers, monetary amounts,times, etc.[17]. Hence, the task of NER is locating and identifying such named entities.

Afaan Oromo is one of a major African languagethatis widely spoken in Ethiopia. The native speakers of the language are the Oromo people, they are the largest ethnic group in Ethiopia,however the exact percentage is controversial. For example according to the researchers Tabor Wami[30], Mohammed Hassen[63] and Tilahun Gamta[70] they comprise about 40 % of the total population of the country, however according to the Ethiopia Population Census Commission 2007 the total population of Oromo People is 34.5% of the total population of the country. Afaan Oromo is also spoken in the neighboring countries like Kenya, Uganda, Tanzania, Djibouti and Somalia. Currently, it is an official language of Oromia Regional State.With regard to the writing system, Qubee(a Latin-based alphabet) has been adopted and become the official script of Afaan Oromo since 1991. Though Afaan Oromo is one of the major languages in Ethiopia, and

spoken by more than 50% of the total population of the country, research works on NLP like NER are scarce. Thus, this research work is concerned with AONER.

1.2. Statement of the Problem

NER is a technology for recognizing proper nouns (entities) in text and associating them with the appropriate types. Some NER systems are incorporated into Parts-of-Speech (POS) taggers, though there are also many stand-alone applications. Whereas most NER systems are based on analyzing patterns of POS tags, they also often make use of lists of typed entities (like list of possible person names) or regular expressions for particular types (like address patterns).

The task has also very great significance in the Internet search engines and in many of the language engineering applications such as Machine Translation, Question-Answering systems, Indexing for Information Retrieval and Automatic Text Summarization [1,11]. These facts show that NERsystem is the basic and relevant component for the development of most of NLP related applications.

Regardless of the above facts, though Afaan Oromo is among the major languages in Africa with a lot of vocabularies, it lacks enough research work on NLP, particularly the NER. As far as the researchers see there is only single research work on Afaan Oromo Named Entity Recognition (AONER) and this was done by statistical approachusing Conditional Random Fields (CRF) algorithm [1]. This approach has its own defect in classifying complex entities.Moreover, it requires large annotated training data to identify named entities. However, the hybrid system applied in this research work easily identifies the complex entities, and it does not require large annotated training data. Accordingly, this study focuses on the effects of resultant NER System for Afaan Oromo by combining both the rule-based and machine learning approaches into a hybrid system so as to increase the efficiency and to identify the complex entities in Afaan Oromo texts.

1.3. Objective of the study

General Objective

The general objective of this study is to develop Afaan Oromo NER system using hybrid approach.

Specific Objectives

- To identify the main aspects and features of named entities in Afaan Oromo.
- To develop rules to identify the Afaan Oromo named entities.
- To integrate the rule based and machine based components of the AONER system.
- To design and develop Afaan Oromo NER system.
- To implement a prototype system to test the findings of this research and evaluate the developedAfaan Oromo NER system.

1.4. Scopeofthe Study

This study is to develop Afaan Oromo NER system. Due to time constraint, the study is limited to recognition of four NEs from different texts in Afaan Oromo:

1. Person Name (PER) – name of a person including his/her father and grandfather.
2. Location name (LOC) – name of geographical entities like cities, regions, country.
3. Organization name (ORG) – name of entities like companies, institutions and organizations.
4. Miscellaneous (MISC) – collects four numeric entities: Date, Time, Monetary values and Percentage.

1.5. Research Methodology

A number of methods (techniques) are employed for the successful completion of this study. Some of them are discussed below.

1.5.1. Corpus collection and Development

Afaan Oromo does not have publicly available annotated corpus text for any NLP task including NER. The researchers collectedAfaan Oromo text data from different documents (such as news paper, text books, research papers, etc.) written in Qubee.Then, the entire NEs in the test corpora are annotated and highlighted manually in addition to the corpora developed byMandefro [1]for

Afaan Oromo.Based on the idea we gathered from Mandefro‟s work andbased on observations, we identified several rules to specify NEs.

Then we developed these rules and implemented using JAVA programming language. Besides, in order to tag NEs in Afaan Oromo texts, lists of trigger words have been identified to recognize the position of proper names in the text. By using keywords we mark noun phrases that might include certain names. Then after,these phrases were processed to extract these names. We have also developed our NER system using GATE system. GATE comes with a default NER systemANNIE[26]. It is the most freely available advanced tool which many researchers are using around the world these days [31]. In addition, in the design and implementation of our system, Afaan Oromo Morphological analyzeris built over GATE to achieve our goals. Without detail knowledge of the language, Computational knowledge is not enough to develop Afaan Oromo NER system.Therefore, linguists and experts in the area of Afaan Oromo have been consulted.

1.5.2. Review of Literature

Different literatures that are considered to be relevant for our research were reviewed and adopted for our work in addition to resources like books, journals and other documents from the Internetfor the purpose of understanding the morphology of Afaan Oromo words. Some of the recent research works particularly in languages such as English, Afaan Oromo, Indian Languages, and others are extensively reviewed in subsequent section.

1.5.3. Development Environment and Tools

To design and develop Afaan Oromo NER, we use windows operating system and GATE development which easily integrated into Statistical method developed by Mandefro [1] using programming language.

1.5.4. Prototype Development

Prototypes was developed for rule based by using GATE developer and integrate into statistical based approach developed by Mandefro [1] using java programming language on LingPipe library system.

1.5.5. Experimental Evaluation

After the prototype was developed, it was tested and evaluated with different test datasets. The performances obtained throughout the conducted experiments were given in three evaluation metrics: Precision, Recall and F-measure.

1.6. Application of Results

NEs are recognized as an important source of information for many applications in NLP.NER is a technology for recognizing proper nouns in text and associating them with the appropriate types. This is the problem of many researchers [1, 32]. The main challenge of this problem is to find the suitable NER systems and try to find the difference of their NE types. NER software serves as an important preprocessing tool for tasks such as information extraction, information retrieval and other text processing applications. Based on these facts, Afaan Oromo NER system plays a crucial role in information extraction based researches and applications associated to the language. It also opens the track for future Afaan Oromo NLP studies. It also simplifies development of the following applications for Afaan Oromo.

- Search Engines (Semantic based)
- Machine Translation
- Question-Answering Systems
- Indexing for Information Retrieval
- Automatic text Summarization

The finding of this study will be used as stepping stone for any researcher to investigate further research on the area.

1.7. Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 detailed literature review on different issues in developing NER. In this chapter, types of NEs, issues related with NER, NLP, IE, GATE and the different techniques and approaches to develop NER system are discussed in detail. Chapter 3 is devoted to discuss related works done for Afaan Oromo and for other languages. Chapter 4 specifies the characteristics of Afaan Oromo illustrating the writing properties of the language and other language specific issues such as the writing system, word formation process

like inflections, derivations and compounding have been extensively presented. Chapter 5 discusses the architecture and design of our system, Afaan Oromo NER. The architectural components of our system along with GATE Developer IDE are briefly discussed in this chapter. The implementation issues are presented in chapter 6. This chapter details stem classification issues, signature building, algorithms and implementation of rules. Chapter 7 is devoted to the evaluation of the system. Chapter 8 concludes the thesis by outlining the benefits obtained from the research work and limitations of the system. It also shows some research directions and recommendations.

Chapter Two: Literature Review

This chapter describes NER and applications of NER. There is a detail discussion of the Characteristics of Afaan Oromo along with the inherent challenges in Natural Language Processing for Afaan Oromo. Different literatures in the field of NER for Afaan Oromo and other languagesare reviewed. Lastly, the concepts that are important for obtaining basic understanding of the ideas on the research area are presented. The review aims to clarify fundamental points about the design and the implementation of AONER.

2.1. Afaan Oromo

Ethiopia is one of the multilingual countries. It constitutes more than 80 ethnic groups [3] with diversified linguistic backgrounds. The country comprises the Afro-Asiatic super family (Cushitic, Semitic, Omotic and Nilotic). Afaan Oromo belongs to an East Cushitic language family of the Afro-Asiatic language super family. It is the most widely spoken language in Ethiopia. It has around 40 million speakers, 50% of the total population of the country, native speakers and the most populous language of Ethiopia [30,62,63,65]. According to Tabor Wami [30] it is also the third most widely spoken language in Africa next to Arabic and Hausa languages. In light of this Gragg[64] also says, more than two-thirds of the speakers of the Cushitic languages are Oromo or speak Afaan Oromo which is also the third largest Afro-Asiatic language in the world [26]. In the Horn of Africa alone, there are over 45 million native Afaan Oromo speakers.

Currently, it is an official language of Oromia regional state. Since 1994 Afaan Oromo is a medium of instruction in primary schools and teacher training institutions of Oromia Regional State. In addition to this, it has been taught as a subject in secondary schools in Oromia Region. From 2003 onwards, however, universities like Jimma and Haramaya started offering BA program in Afaan Oromo and Literature. The demand for qualified researchers in Afaan Oromo Language and Literature is also increasing from time to time due to the fact that the language has become the official language in Oromia Regional State, and it is a medium of communication in some media institution of the Federal Government as well.

2.1.1. Writing system of Afaan Oromo

The writing system of Afaan Oromo is nearly phonetic since it is written the way it spoken, i.e. one letter corresponds to one sound. The language usesLatin alphabet (Qubee) which was formally adopted in 1991[65], and it has its own consonants and vowels sounds. Afaan Oromo has thirty-three consonants, of these seven of them are combined consonant letters: **ch, dh, ny, ph, sh, ts** and **zh**.. The combined consonant letters are known as **„qubee dachaa'**. Afaan Oromo has five short and five long vowels. The Afaan Oromo alphabet is characterized by capital and small letters like English alphabet. In Afaan Oromo, as in English language, vowels are sound makers and do stand by themselves.

2.1.2. Word Categories of Afaan Oromo

Words are the basic part of a given language. The arrangement of word or their combination depends on the rule or grammar of that language. The combination of these words on the bases of the language gives us sentences. The meanings of these sentences depend on each word of the sentence and the way they are arranged. However, the extent to which a given word determines the meaning of a sentence depends on the contribution of that word. All words do not have equal contributions to sentence meaning. Their contribution depends on their category and their feature. Based on the category of the word; we can find out the contribution of that word. In addition to this, we can easily identify the rules to be applied to this word in case of morphological change [66]. In general, word categories can be identified by looking at the meaning (semantic) of that word, by looking at the form (morphology) of that word or by looking at the actual position (syntax) of that word [66].

The grammatical categories of Afaan Oromo have undergone a series of improvement in terms of its word categories and other syntactic features. Traditionally, Afaan Oromo words are categorized into eight grammatical categories (Noun, Verb, Adjective, Adverb, Adposition, Pronoun, Conjunction and Interjection). But according to some researchers Mandefro Legesse, Assefa W/mariam and Getachew Mamo [1,3,69] Afaan Oromo has five grammatical categories nouns, verbs, adverbs, adjectives and adposition. According to these researchers pronouns included under the noun category, and conjunctions and interjections under adposition. From this point one can

understand that Afaan Oromo has five major grammatical categories serving as heads in phrase construction.

Each of these classes again can be divided into other sub-classes. For instance, noun class is categorized as proper noun, common noun and pronoun, and Preposition and postpositions are sub classes of ad- positions. The subclasses in turn can be divided into subclasses, and the subdivision process may continue iteratively depending on the level and aim of the investigation. We will see some of Afaan Oromo word categories that have contributions to our study which was analyzed in the Mandefroʺs research work [1].

Noun

Nouns are names that are used to name or identify things, people, animals, places or abstract ideas. In Afaan Oromo most of the time a sentence begins with a noun which starts with capital letter and it uses a noun as a subject followed with subject markers (-ni, or -n,). Direct object and indirect object also optionally follows the noun which is the subject of a sentence. Every word that has affixes like –eenya, -ina,-umma are nouns in Afaan Oromo. The bolded words in the following sentences are all nouns:

Hoolaanmarga dheeda. (The sheep grazes grass).

Namnikamiyu niduʺa. (Man is mortal).

Bunnidinaggee keeynaa gudisa. (Coffee develops our economy).

Najjoon godina Wallaga Lixaa keessatti argamti. (Nejo is found in West Wollega zone).

Verb

Verb is the most important part of a sentence that says something about the subject of a sentence, expresses an actions, events or states of being. In Afaan Oromo verb occurs in the final positions of a sentence. It is not the case that verbs constitute a distinct, open word class in all languages. In Afaan Oromo verbs are forms which occur in clause final positions and belong to a distinct category [1]. In each of the following sentences the verb is bolded:

Caalaan farda **bite**. (Chala bought a horse)

Caaltuun barattuu **dha**. (Chaltu is a student)

Leensaan **dhufte**. (Lensa has come)

Adverb

Adverbs are words which are used to modify a verb, an adjective, another adverb, or a clause. Adverbs usually precede the verbs they modify or describe. An adverb indicates time, manner, place, cause, or degree and answers questions such as „how?", „when?", „where?", and „how much?". In the following examples, each of the bold words is an adverb:

Oboleessi koo**boru** deema. (My brother will leave tomorrow.) Boru (tomorrow) is an adverb.

Namichi **tasa**du"e . (The man accidentally died.) Tasa(accidentally) is an adverb.

Adjective

An adjective modifies a noun or a pronoun by describing, identifying, or quantifying words. In Afaan Oromo an adjective usually follows the noun or the pronoun which it modifies. Some of Afaan Oromo adjectives are: **hedduu**, **mara**, **kam**, **adii**, **qalla**, **tokko**, **kee**, etc. In the following examples, the bold words are adjectives:

Gammachuun **qalla** dha. (Gemechu is thin).

Konkolaatan Cala **adii** dha. (Chala"s car is white).

Pronoun

Alike English, in Afaan Oromo **pronoun** can replace a noun or another pronoun. Pronouns are marked for number and gender. For example, pronouns like "ishee/isii" which means “she” is feminine (singular), "isa" which means 'he' is masculine (singular), "isaan” which means 'they' is plural and can be masculine or feminine and "nuyi" which means “we” is plural and can be masculine or feminine. We use pronouns to make sentences less cumbersome and less repetitive .

Grammarians classify pronouns based on their functions and meanings in the sentence into several types, including the personal pronoun, the demonstrative pronoun, the interrogative pronoun, the indefinite pronoun, the relative pronoun, the reflexive pronoun, and the intensive pronoun [1,69].

Table 2.1: Personal Pronouns of Afaan Oromo

	1st person	2nd person	3rd person
singular	Ani(I)	Ati(you)	Isa/inni(he) Isii/ishee(she)
plural	Nuti(We)	Isin(you)	Isaan/Jarri(they)

For example:

Tolan dhufe. (Tola came.)

Inni dhufe. (He came.)

Adpositions

Adpositions are traditionally defined as words that link to other words, phrases, and clauses and express spatial or temporal relations. Adpositions are almost universal part of speech. It is a cover term for prepositions and postpositions. It is a member of a closed set of items that occurs before or after a complement composed of a noun phrase, noun, pronoun, or clause that functions as a

noun phrase, and form a single structure with the complement to express its grammatical and semantic relation to another unit within a clause[67,68].

Some languages have either prepositions or postpositions, others have both and yet others have quite neither. For example English has prepositions but Japanese has postpositions [67, 68]. Unlike English and Japanese, Afaan Oromo has both prepositions and postpositions. As grammatical tools, adposition marks the relationship between two parts of a sentence: characteristically one element governs a noun or noun-like word or phrase while the other functions as a predicate.

Qotebula-**dhaaf** (for farmers) Ambo-**tti**(at Ambo) Saartuu-**n** (Sartu is)

Numaa-**f**(for us) Gaara-**rraa**(from above)

Adpositions are traditionally defined as words that "link to other words, phrases, and clauses" and that "express spatial or temporal relations."

2.1.3. Named Entities in Afaan Oromo

Named entity is a term or phrase that identifies an object from a set of other objects with in similar attributes. As stated before, it is also known as entity identifying and entity classifying. It is used to name a person, organization, location and other things. In this section, we will discuss the nature of NEs in Afaan Oromo based on Mandefro"s [1] research work. As described earlier, our study focused on the recognition of the following types of NEs: Person, Location, Organization and Miscellaneous. The miscellaneous category includes Date, Time, Monetary Value and Percentage. The discussions made here are based on the pattern of NEs noticed during corpus development by combing corpus developed by Mandefro [1] and from the general background of the researcher. We discuss different methods that help the researchers to identify NEs from Afaan Oromo texts based on [1] documents and detailed discussions with Afaan Oromo linguists.

Numerals in Afaan Oromo

Numerals are words representing numbers or quantity of something [1]. They can be cardinal or ordinal numbers. Ordinal nouns in Afaan Oromo are formed by adding suffix" **–ffaa**" to the cardinal numbers.

For example:

Cardinal	English	Ordinal	English
shan	five	shan-affaa	fifth
sadii	three	sad-affa	third
lama	two	lamm-affaa	second

Like English, Afaan Oromo has distribution numerals, compound numerals and special numerals. The following are examples to illustrate this.

Example **lama lama** ("two two") - distribution numeral

dhiba shan ("five hundred") - compound numeral

Dhiba shan-affaa ("five hundredths") - compound numeral

Dhiba shan-fi afur("five hundred and four) - compound numeral

walakkaa ("half") - special numeral

Date/Time in Afaan Oromo

In Afaan Oromo time expressions could be composed of a decimal or cardinal number followed by a word or expression representing a time measure (e.g. **sa'a afour** "four O'clock").

The most common expressions identifying points or stretches of the time line are within the scope of the time type. There are three subtypes proved to be adequate for the distinction between the expressions in question:

- Date: can be written by cardinal or number. It is an expression representing a date, whose components can be a day of the week (e.g. **Jimaata**), a day of the month (e.g. 28), a month (e.g. **Hagayya**) or a year (e.g. **Bara 2014**).
- Time periods: It is an expression written by cardinal or numbers and an explicit indication of a period of time concerning a specific year, decade or century (e.g. **jaarra 1**).
- Time of the day: expressions with different formats, indicating a specific time of the day (e.g**. sa'atii 11**).

Monetary Values in Afaan Oromo

Monetary values or Currency can be written either in decimal or cardinal number followed by a word or expression representing a currency.

e.g.**qarshii 2**(“2 birr) or **Qarshii lama**(two birr)

Percentages in Afaan Oromo

A percent is part of something, split into a hundred pieces. It is calculated by comparing a part of something compared to the whole. . A percent can go from 0 percent to 100 percent and it is a very handy way of writing fractions. Thus, Percentages can be compared more easily than fractions. The symbol to represent a percentage, %, is also used in Afaan Oromo. But here it comes before the number, unlike that of English. The word „percent“ has also an equivalent in Afaan Oromo as „**dhibbantaa’** or „**dhibbeentaa’**. Example:

Dhibbeentaa 22n dabale. (22 percent increase).

%2ndabale. (2% increased).

2.1.4. Syntax of Afaan Oromo

In all natural languages, there is a standardized word order in a sentence. For example, English and French languages have word orders of Subject -Verb- Object order. Afaan Oromo and English have differences in their syntactic structure. Afaan Oromo uses subject-object-verb (SOV) form which is similar to Amharic and Japanese languages [69]. Subject-verb-object (SOV) is a sentence structure where the subject comes first, and the object and the verb are second and third elements of a sentence respectively. For instance, in the Afaan Oromo sentence **Qananisaan atileeti dha** (Kenenisa is an athlete), **Qananiisa** (kenenisa)is a subject, **atileeti** (athlete) is an object and **dha** (is) is a verb.

Afaan Oromo adjectives follow a noun or pronoun; their normal position is close to the noun they modify while in English adjectives usually precede the noun. For instance, **nama cima** (strong man), the adjective **cima** (strong)follows the noun **nama** (man).

There are different rules to word order in Afaan Oromo sentence construction understanding of this syntactic structure of sentence can help us to know the relationship between words which in turn leads us to categorize them correctly. For our work we mainly focus on Main clause word order and Noun phrase word order. The points discussed in this section are based on Mandefro [1]

Main clause word order

As we discussed Afaan Oromo uses subject-object-verb (SOV) format sentence construction in the main clause. But, the resulting sentence structure may vary as shown in the following cases.

I. Subject + verb

Daraartuun dhufte.(Derartu came.)

Anaa dhufu. (Welcome.)

II. Subject + complement (object or adverbial) + verb

Calaan konkolaataan dhufe .(Chela came by car.)

Beekaan mana baruumsaati dhufe. (Beka came from school.)

III. Complement + verb

Mirga ishee falmatte. (She defined her right.)

Tokkummaa qabaadhaa. (Have a unity).

Noun phrase word order

The noun usually precedes the qualifiers. The qualifiers are arranged in the sequence: noun – adjective – possessive/demonstrative pronoun. The subject marker is added to the noun itself and to the qualifiers of the noun as the base form is repeated in all qualifiers of the noun.

Odaan guddaan kun yoom dhaabbate? (When this Sycamore tree is planted?)

Mucaan qaloon xiqoon kun cimtuudha. (This small thin boy is dynamic.)

2.2. Natural Language Processing

Natural language processing (NLP) is an interdisciplinary area based on many fields of study, which is used for designing and building software that can analyze, understand, and generate natural language. Some of the tasks of NLP that provides a potential means of gaining access to the information inherent in the large amount of text are IR and IE. Information retrieval systems typically allow a user to retrieve documents from a large database. During the information retrieval process a user expresses an information need through a query. IE is the task of automatically extracting structured information from unstructured or semi-structured documents.

NLP is a computational methodthat automates the translation process between computer and human languages. It is a method of getting a computer to understandably read a line of text without the computer being fed some sort of clue or calculation. The goal is to enable natural languages, such as English, French, Afaan Oromo and others to serve either as the medium through which users interact with computer systems as such as database management systems and expert systems (natural language interaction), or as the object that a system processes into some more use full form such as in automatic text translation or text summarization (natural language text processing).[13]. It can also be defined as an area of active research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things [14]. It helps to extract interesting, non-trivial knowledge from free or unstructured text, and this can be put roughly as figuring out who did what to whom, when, where, how and why. NLP typically makes use of linguistic concepts such as part-of-speech (noun, verb, adjective, etc.) and grammatical structure.

NLP researchers aim to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform the desired tasks [33]. This is based on both a set of theories and a set of technologies.In recent years, the natural language text interpretation and processing technologies have also gained an increasing level of sophistication. NLP technologies are becoming extremely important in the creation of user-friendly decision-support systems for everyday non-expert users, particularly in the areas of knowledge acquisition, information retrieval and language translation[31].

NLP is widely used in modern computer systems. It is concerned with the interactions between computers and human languages. As such, NLP is related to the area of human-computer interaction. Many challenges in NLP involves natural(human) language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation.

The goal of natural language processing (NLP) is to build computational models of natural language for its analysis and generation. There are more practical goals for NLP; many of which are related to the particular application for which it is being utilized. For example, technological motivation of building intelligent computer systems such as machine translation systems, speech understanding systems, text analysis and understanding systems, computer aided instruction systems, and a cognitive and linguistic motivation to gain a better insight into how humans communicate using natural language [23].

Currently, the field of NLP includes a wide variety of linguistic theories, cognitive models, andengineering approaches. Computer science is one of these fields, which provides techniques for model representation, and algorithm design and implementation. These models classified words not only based on meaning, but also on their co-occurrence with other words.

Although unrestricted NLP is still a very complex problem (and accordingto some, an AI-complete problem), numerous successful systems exist for restricted domains ofdiscourse[12].The advances in computer science which have made NLP possible today are bigger and faster computers, the development of major software systems and the development of database technology. They assist the user in writing computer programs (such as parsers) or preparing the data (such as grammars and lexicon related data). The most important among them are high-level languages and operating systems. Developing database systems allows easy storage on and retrieval from disk of large dictionaries, lexicons, lexicalized grammars, and corpora of text.

2.3. Information Extraction

Information extraction (IE) is a subfieldof natural language processing that is concerned with identifying predefined types of information from text by applying natural language processing. In most of the cases, this activity is concerned with processing human language texts by means of

natural language processing (NLP) [26]. In other words, the goal of an IE system is to find and link the relevant information while ignoring the extraneous and irrelevant one [16].For example, an information extraction system designed for a business domain might extract the names of companies, products, facilities, and financial figures associated with business activities. Another example is that recent activities in multimedia document processing like automatic annotation and content extraction out of images, audio or video could be seen as IE.

Automatic extraction of entities, relationships between entities, and attributes describing entities from data is the major goal of IE. These entities enable much richer forms of queries on the abundant unstructured sources than possible with keyword searches alone. A more specific goal is to allow logical reasoning to draw inferences based on the logical content of the input data. Structured data is semantically well-defined data from a chosen target domain, interpreted with respect to category and context.

In general, information extraction (IE) consists of two main subtasks: the detection of the interested NE and the extraction of information for the specific named entity. The extracted information can be used for a number of purposes, for example to prepare a summary of texts, to populate databases, fill-in slots in frames, identify keywords and phrase for information retrieval, and so on. IE techniques are also used for classifying text items according to some pre-defined categories [15,22,24].

The development of IE technology is closely bound to Message Understanding Conferences (MUC), which took place from1987until 1998[28]. One of the most ambitious forms of information extraction is event extraction which involves retrieving a wide variety of information about a certain type of event from a document. This task was one of the focal points of the Message Understanding Conferences (MUC) over the years, particularly MUCs 3 through 5 (1991-1993). For instance, the MUC-3 and MUC-4 event extraction task concentrated on finding details of terrorist attacks in newswire documents [34] such as date, location, target, instrument and actor. The MUC efforts, among others, have consolidated IE as a useful technology for Text Based Intelligent systems.

There is also development of modern systems on IE not only marking relevant documents but also identify, extract and present relevant and interesting content. Since information extraction relies for the most part on a statistical approach to NLP, there is no real understanding of the text

involved for the most part of it is simply a matter of recognizing linguistic patterns. Indeed, research has shown that shallow parsing is sufficient for tasks such as information extraction or question answering [30]: complete syntactic analyses are often not necessary for acceptable performance on these kinds of tasks. However, a real understanding of texts is still required for the best possible solutions to these kinds of tasks.

2.3.1. Architecture of Information Extraction System

There are four basic components in the main modules of IE systems, which are tokenization, morphological and lexical processing, syntactic analysis, and domain analysis [22].The factors which will affect the system are summarized by Piskorski and Yangarber [15] whether it requires additional components over the four essential components as follows:

- Language of the text: Some languages like Semitic Languages will require morphological and word segmentation processing.
- Genre: For instance, informal text may contain misspellings and ungrammatical constructs that require special analysis that newspaper texts in general do not need.
- Text properties: Very long texts may require IR techniques to identify the relevant sections for processing.
- Task: Tasks like entity identification are relatively simple, but if the task involves extracting events, then entire clauses may have to be analyzed together.

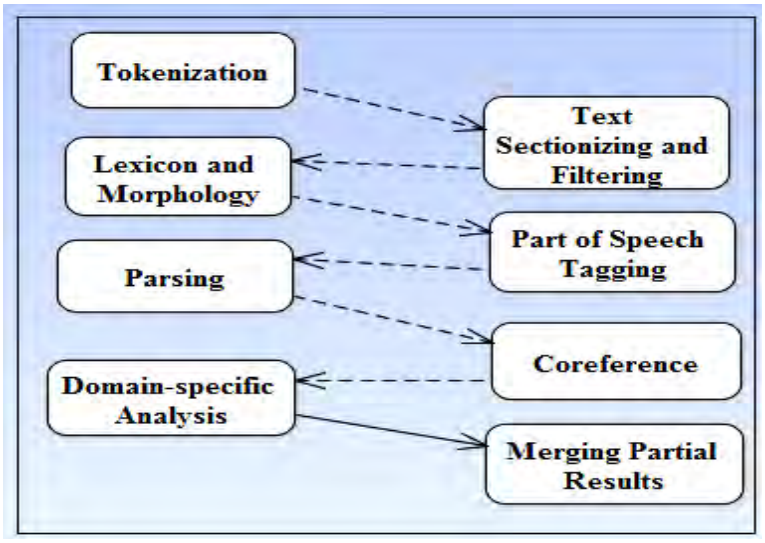

Figure 2.1: The architecture of information extraction systems [22]

Figure 2.1shows the architecture for a simple information extraction system. It begins by processing a document using several procedures. First, the raw text of the document is split into sentences using a sentence segmenter, and each sentence is further subdivided into words using a tokeniser. Next, each sentence is tagged with part-of-speech tags, and this step is helpful in NE detection which is the third step. In the third step, we search for mentions of potentially interesting entities in each sentence. Finally, we use relation detection to search for likely relations between different entities in the text.

2.3.2. Types of Information Extraction (IE)

Applying IE on text aims at creating a structured view, which is representation of the information that is machine understandable. The classic IE tasks performed by IE systems usually differ, but the following classification from the seventh Message Understanding Conference [6] characterizes the most common tasks:

- NER:addresses the problem of the identification and classification of predefined types of NEs, such as organizations (e.g., "World Health Organisation"), persons (e.g., "Abebe Bikila"), place names (e.g., „Meddawolabu"), temporal expressions (e.g., „1 September 2011"), numerical and currency expressions (e.g., „20 Million Euros"), etc.
- Coreference Resolution (CO) –refers to the identification of multiple(coreferring) mentions of the same entity in the text.
- Relation Extraction (RE) - is the task of detecting and classifying predefined relationships between entities identified in text.
- Event Extraction (EE) - refers to the task of identifying events in free text and deriving detailed and structured information about them, ideally identifying who did what to whom, when, where, through what methods,and why. Usually, event extraction involves extraction of several entities and relationships between them.

2.4. Named Entities

NE is a word or a phrase that clearly identifies one item from a set of other items that have similar attributes. Most NEs are proper nouns; like name of a person, an organization, and locations. For example, the sentence "The American secretarial general John Carey visited Ethiopia" contains three NEs: America and Ethiopia are a country name and Johan Carey is a person name.

Moreover, NE tasks often include expressions for date and time, names of sports and adventure activities.

The concept of NE was introduced in the sixth Message Understanding Conference, which is currently used in IE technology [8,35].In fact, the MUC conferences were the events that have contributed in a decisive way to the research of this area. It has provided the benchmark for named entity systems that performed a variety of information extraction tasks [1].

MUC 7classifies named entities into the following categories and subcategories [6]:

1. Entity (ENAMEX): person, organization, location
2. Time expression (TIMEX): date, time.
3. Numeric expression (NUMEX): money, percent.

Named entities can consist of any type of word such as adverbs, prepositions, adjectives, and some verbs, but the majority of named entities are made up of nouns [26].

2.5. Named Entity Recognition

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.This in turn means that every word needs to be categorized as belonging to a named entity or not.

NER is one of the most commonly usedtype of IE [6,11], where structured text is extracted from unstructured text, such as newspaper articles which substantially helps several other NLP tasks. In other words, we can say that itis a standard component of IE, enabling the extraction of useful information from documents[23]. It is also a key component when integrated into technologies that deal with large amounts of information, such as, IE, IR, Machine Translation (MT), Speech Recognition and Question Answering (QA) tasks [15].

Different Message Understanding Conference were conducted in different times to improve the automated text extraction, which undertake measurements to determine the capability of software systems to correctly identify and extract information from text documents. For example, in MUC6 there has been increasing interest in NER and extensive effort has been devoted into research.

Major computational linguistic conferences hosted special tracks for the task and there has been steady growth of publications throughout the years. Several events made the attempt to enrich the definition of the task [75].In addition, during MUC-7 conference in 1998, date and time entities, and multi-lingual named entity recognition was introduced and evaluation of Named Entity Recognition task, a co-reference task, and three tasks involving extracting information into templates undertaken [6].

The Automatic Content Extraction (ACE) program introduced several new entity types and a more fine-grained structure of entity sub-types in an attempt to achieve more precise classification of entities, such as distinguishing government, educational and commercial organizations from each other, which all belong to the coarse-grained entity type organization [37]. From the above points we understand that there is improvement of NER system from time to time on different languages.

2.5.1. Named Entity types

Named Entity Recognition (NER) is a type of information extraction that seeks to identify regions of text corresponding to named entities and to categorize them into a predefined list of entity types. Names of persons, locations, and organizations are the most studied entity types since MUC-6. The majority of previous researches performed on NER. For example, MUC-7 by Hirschman and Chinchor has focused on those entity types[38]. CoNLL-2002 and 2003 added a miscellaneous type [39], and ACE introduced geo-political entities, weapons, vehicles and facilities [37]. Other examples include Ontonotes‟ categorization into 18 classes [41] and BBN‟s 29 answer types [42] While these representations are more expressive than the person-location- organization standard, they still fail to distinguish entity classes which are common when extracting hundreds or thousands of different relations [4,43]. Some work does not limit the possible types to extract and is referred as open domain NER [18,44]. For example [45] defined a named entities types, which includes many atomic subcategories, such as international organization, river, or airport, and adds a wide range of categories, such as product, event, substance, animal, religion, or color.
Finally, there are no general types of NE that are commonly used across all languages. As a result of this, the number and type of NEs in NER system can recognize differs from language to language or from domain to domain.

2.5.2. Architecture of NER System

NER system has four basic components regardless of whether it is designed according to rule-based approach or automatic machine learning approach.

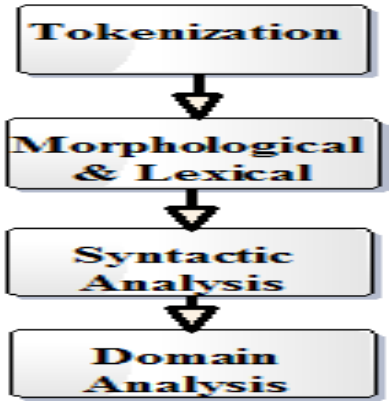

Figure 2.2: Architecture of a typical Information Extraction System

The core elements are Tokenization, Morphological and Lexical processing, identification andClassification [33,46] Tokenization is the first step in interpreting text by splitting up a string of words/characters (comprising a document, paragraph or sentence) into minimal parts of structured text that are useful to be used as a unit, referred to as a token [33[With regard to NER, tokenization can consist of sentence splitting and word segmentation as a subtask. After tokenization process is completed, morphological and lexical processing proceeds. During this step, word tokens in a document are sequentially tagged as being inside or Outside of a given named entity. It mainly employs part -of-speech tagger [46] and each word in a sequence of words is labeled with an Inside or Outside tag [47]. In addition, it employs components like NP chunking and feature extraction. The morphological and lexical processing mainly helps for the detection of NEs which is depicted in the Figure 2.2 as identification. The identification component detects NEs with the aid of stored models or rules, based on the approach used. The detected NEs are then ready to be classified into their respective classes. The classification step takes the detected NEs andcategorizes them into their corresponding categories. The classification is done by a classifier.

2.5.3. Named Entity Recognition Approaches

The basic aim of NER is to extract and classify names into some particular categories from given corpus or text. In order to do this one has to apply the NER approaches. These approaches for

recognizing named entities from text have been divided into three categories: Rule based NER, Machine learning based NER and Hybrid NER. Each approach is briefly discussed as follows.

Rule-based approaches

In Rule based approach, rules are developed to identify NE in text. This approach takes muchtime in development and one should have good knowledge of target language. This approach relies on handcrafted rules that require strong linguistic skills and manually compiled corpora and produce better results for restricted domains [1]. These kinds of systems have better results for restricted domains and are capable of detecting complex entities that are difficult with learning models. The accuracies of rule-based systems are considered as gold standard as they reach near to perfection. The main advantage associated to hand-crafted rule based systems is that extraction logic of complex entities can be fine tuned and it does not require a large amount of pre-annotated data or corpus.

In addition, rule-based approach is domain and language specific; therefore, it has the problem of customizing or adapting the system with new domains or new languages. Also creation, modification and maintenance of handcrafted rules and lexicons are a time consuming process and has dependency on the availability of a domain expert. Moreover, the grammar-based technique requires linguistic expertise and strenuous efforts to build a NER system for every new language [32].

Statistical approach

Statistical approach is also known as Machine Learning approach. This is a swift approach to develop a NER system.In Machine Learning-based methods, system looks for patterns and relationships in text to make a model using statistical techniques and machine learning algorithms. This approach utilizes large corpora and features extracted from datasets annotated with NEs, to train a classifier to recognize a named entity[21]. On the other hand, machine learning approaches need a training dataset which is tagged in a certain manner to recognize new entities from new testing dataset of the same domain. Besides a precise selection of features is required [8,16,48]. Hence, this approach converts the NE recognition task into a classification task. Nowadays, machine-learning approaches are popularly used in NER because these are easily trainable; adaptable to different domains and languages, and are easily maintainable [49].

There are three types of machine learning methods: namely Supervised, Semi-supervised and Unsupervised. A Supervised Learning method performs tagging of words of a test corpus when they are annotated as entities in the training corpus. Supervised learning require large amount of training data for good performance. Therefore, it learns the relationship between clues defined by the selected features and the correct answers provided by the training examples [50].The main technique for Semi-supervised learning is bootstrapping and involves a small degree of supervision such as a set of seeds for starting the learning process. On the other hand, unsupervised machine learning methods do not rely on training examples, but instead use a clustering algorithm that groups together similar text objects based on the feature set and are useful when the features have a large variation [60]. Unsupervised learning is also required when the classes are not known beforehand, when training examples are not available or the cost to produce training examples is too high, or when objects and features change dynamically. In this method, named entities can be gathered from clustered groups based on the similarity of context. The Unsupervised learning techniques rely on lexical resources, lexical patterns and statistics computed on a large annotated corpus. This approach can be easily ported to different domain or languages. Supervised machine learning methods for IE have proven to be more popular than the other two methods in part due to its better performance [24,46,51].

Hybrid approach

The hybrid approachis the combination of hand crafted rule based system and Machine Learning system.To develop the Hybrid system we use Statistical tools as well as linguistic rules. Combinations of both approaches make a system more accurate and efficient. It takes strongest points from each method to improve the overall performance of the system.

2.5.4. Feature Space for NER

Features are characteristic attributes of words designed for algorithmic consumption [52]. Several steps have to take place before the named entity recognition process can proceed successfully, irrespective of whether it is split up into two phases or combined in a single phase. First, example data needs to be collected, cleaned and prepared before it can be split into training and test data. The first step that the participants themselves have to take is transforming the data into a format appropriate for their system. This is necessary for both handcrafted as well as machine learning

approaches and does not only involve cleaning up the data but also extracting the desired features from the data that are necessary for the recognition process. In many real-world situations, including NLP problems, relevant features are often unknown a priori. Therefore, a large number of predictive, candidate features are introduced to better represent the domain. The problem of determining which features have the greatest predictive value is called feature selection. Feature selection is a set of candidate features; select a subset that performs the best on unseen examples using some classification system. Performing feature selection can have several advantages:

- It reduces the cost of recognition and the execution time by reducing the number of features that need to be collected.
- In some cases feature selection can also provide a better generalization performance due to the exclusion of irrelevant or redundant variables.
- Knowing which features are relevant can give insight to the nature of the problem at hand.

In the following section, we will briefly discuss the threebasic featuresto recognize NE: Word-level features, List lookup features and Document and corpus features which are discussed in detail inMandefro‟s research work [1].

2.5.4.1. Word-level features

Word-level features are related to the character makeup of words. They specifically describe word case, punctuation, numerical value and special characters.

- **Digit Pattern**

Digits can express a wide range of useful information such as dates, percentages, intervals, identifiers, etc. Special attention must be given to some particular patterns of digits. For example, two-digit and four-digit numbers can stand for years [25] and when followed by an “s”, they can stand for a decade; one and two digits may stand for a day or a month [53].

- **Common word ending**

Morphological features are essentially related to words affixes and roots. For instance, a system may learn that a human profession often ends in “ist” (journalist, cyclist) or that nationality and

languages often ends in ”ish” and “an” (Spanish, Danish, Romanian). Another example of common word ending is organization names that often end in “ex”, “tech”, and “soft” [54].

- **Functions over word**

Features can be extracted by applying functions over words. An example is given by Collins and Singer[6] who create a feature by isolating the non-alphabetic characters of a word (e.g., nonalpha(A.T.&T.) = ..&.). Another example is given by Patrick et al [55] who use character n-grams as features.

- **Patterns and summarized patterns**

Pattern features were introduced byBick Eckhard[58] and then used by other researchers. Their role is to map words onto a small set of patterns over character types. For instance, a pattern feature might map all uppercaseletters to “A”, all lowercase letters to “a”, all digits to “0” and all punctuation to “-”:

2.5.4.2. List lookup features

Lists are the privileged features in NER. The terms “gazetteer”, “lexicon” and “dictionary” are often used interchangeably with the term “list”. List inclusion is a way to express the relation “is a” (e.g., London is a city). It may appear obvious that if a word (London) is an element of a list of cities, then the probability of this word to be city, in a given text, is high. However, because of word polysemy, the probability is almost never 1 (e.g., the probability of “Fast” to represent a company is low because of the common adjective “fast” that is much more frequent).Table 2.2 shows some of the list lookup features of NER.

Table 2.2: List lookup features adopted from Mandefro [1]

Features	**Examples**
General Dictionary	- General dictionary - Stop words (function words) - Capitalized nouns (e.g., April, Africa, Thursday) - Common abbreviations
Gazetteers	- Organization, government, airline, educational - First name, last name, celebrity

	- Astral body, continent, country, state, city
List of entity cues	- Typical words in organization. (e.g., “associates”) - Person title, name prefix, post-nominal letters - Location typical word, cardinal point

Most approaches implicitly require candidate words to exactly match at least one element of a pre-existing list. At least three alternate lookup strategies are used in the NER field.

First, words can be stemmed (stripping off both inflectional and derivational suffixes) or lemmatized (normalizing for inflections only) before they are matched [56]. For instance, if a list of cue words contains “technology”, the inflected form “technologies” will be considered as a successful match. For some languages,diacritics can be replaced by their canonical equivalent (e.g., „é‟ replaced by „e‟) [57].

Second, candidate words can be “fuzzy-matched” against the reference list using some kind of threshold edit-distance [51]. This allows capturing small lexical variations in words that are not necessarily derivational or inflectional. For example, Frederick could match Frederik because the edit-distance between the two words is very small (suppression of just one character, the „c‟).

Third, the reference list can be accessed using the Soundex algorithm [59] which normalizes candidate words to their respective Soundex codes. This code is a combination of the first letter of a word plus a three digit code that represents its phonetic sound. Hence, similar sounding names like Lewinskey (soundex =l520) and Lewinsky (soundex = l520) are equivalent in respect to their Soundex code.

2.5.4.3. Document and Corpus features

Document features are defined over both document content and document structure. Large collections of documents (corpora) are also excellent sources of features. Welist in this section features that go beyond the single word and multi-word expression and include meta-information about documents and corpus statistics,document and corpus features are listed in Table 2.3.

- Multiple occurrences and multiple casing
- Entity co-reference and alias
- Document meta-information

- Statistics for Multiword units

Table 2.3: Document and corpus features

Features	Examples
Multiple occurrences	- Other entities in the context - Uppercased and lowercased occurrences - Anaphora, co-reference
Local syntax	- Enumeration, apposition -Positioninsentence, in paragraph, andindocument
Meta information	- Uri, Email header, XML section - Bulleted/numbered lists, tables, figures
Corpus frequency	- Word and phrase frequency - Co-occurrences - Multiword unit permanency

2.6. Tools for AONER

To the best of our knowledge, there is only single NER system for Afaan Oromo which has been developed by Mandefro[1] using Machine learning based approach by using CRF algorithm. This system is capable of identifying Person, Location, Organization and miscellaneous NEs.

On the contrary, we have employed a hybrid approach, to extract named entities from Afaan Oromo texts. The researchers adopted the statistical component of this hybrid approach which is developed by Mandefro[1], and we have developed the rule based component for AONER system using the GATE tool. The details about GATE tool and Weka, which we have used to develop a rule based component and ML component respectively for AONER is discussed in the subsequent section.

2.6.1. GATE

GATE is a framework and architecture for developing and deploying software component that process human language. As a framework, it provides reusable implementations for language processing components and a set of software building blocks that researchers can use, extend and customise for their specific needs.

As a development environment, it has a set of standardised interfaces to reusable components and provides a variety of data visualisation, debugging and evaluation tools that minimise the time researchers spend in building new language processing systems or modifying existing ones, by aiding overall development and providing a debugging mechanism for new modules. Because GATE has a component-based model, this allows for easy coupling and decoupling of the processors, there by facilitating comparison of alternative configurations of the system or different implementations of the same module (e.g., different parsers). The availability of tools for easy visualisation of data at each point during the development process aids immediate interpretation of the results [48].

The environment has facilities to view documents, corpora [60], and linguistic data (expressed as annotations), e.g., Figure 2.2 shows the document viewer with some annotations highlighted. It also shows the resource panel on the left with all loaded applications, language resources, and processing resources (i.e., modules). There are also viewers/editors for complex linguistic data like co-reference chains (Figure 2.3) and syntax trees (Figure 2.4). New graphical components can be integrated easily, thus allowing researchers to customise the environment as necessary.

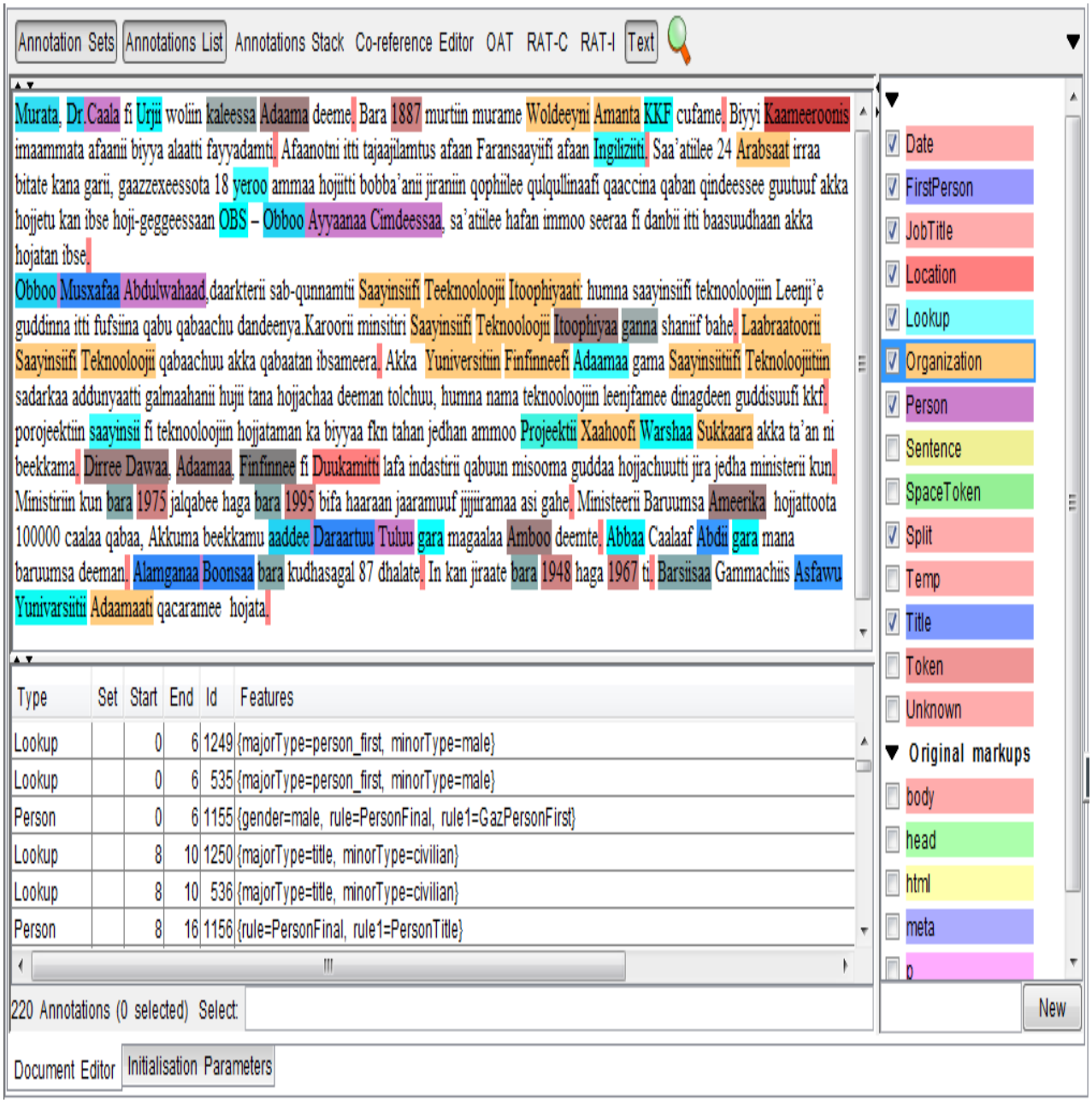

Figure 2.3: Screenshot of GATE

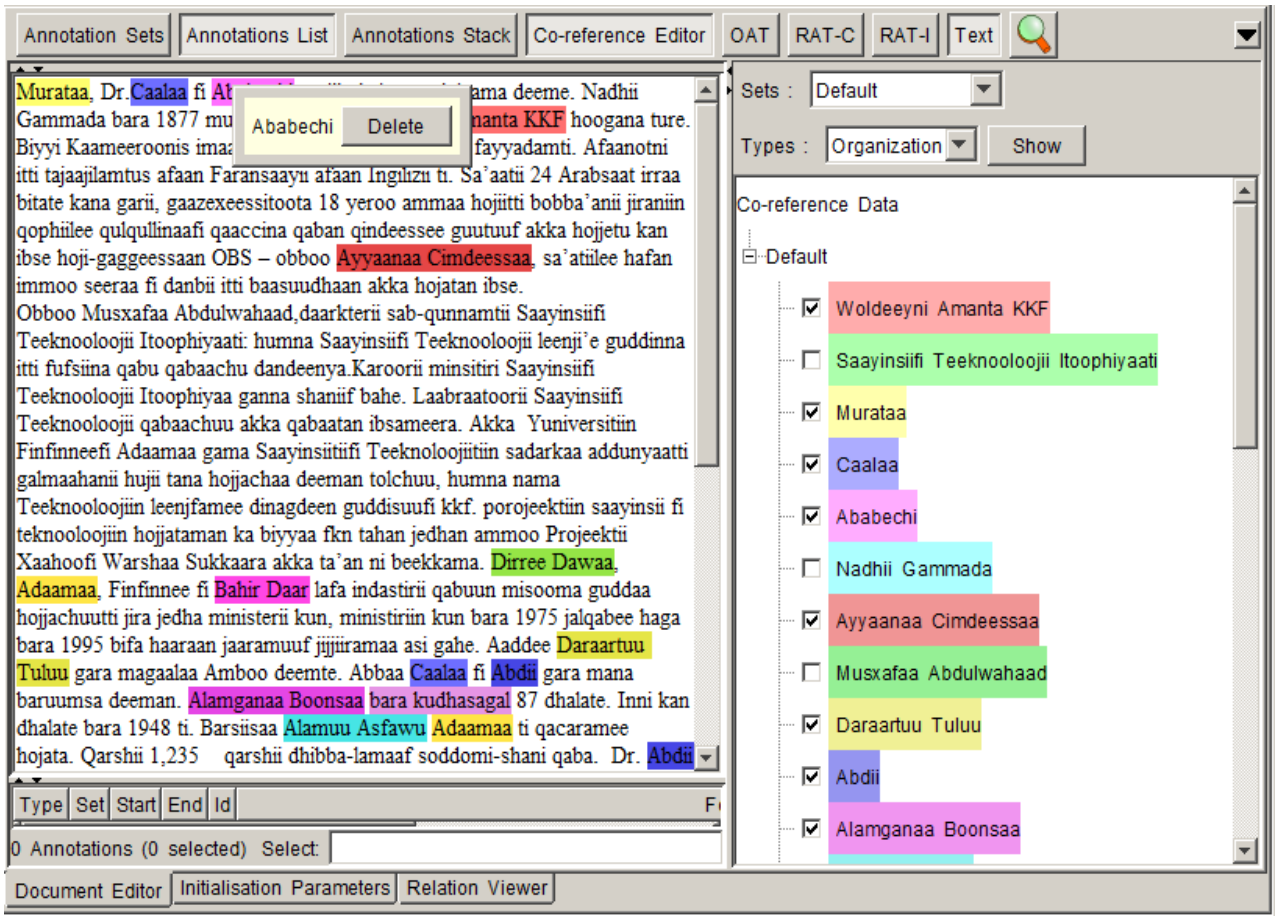

Figure 2.4: The coreference chains viewer

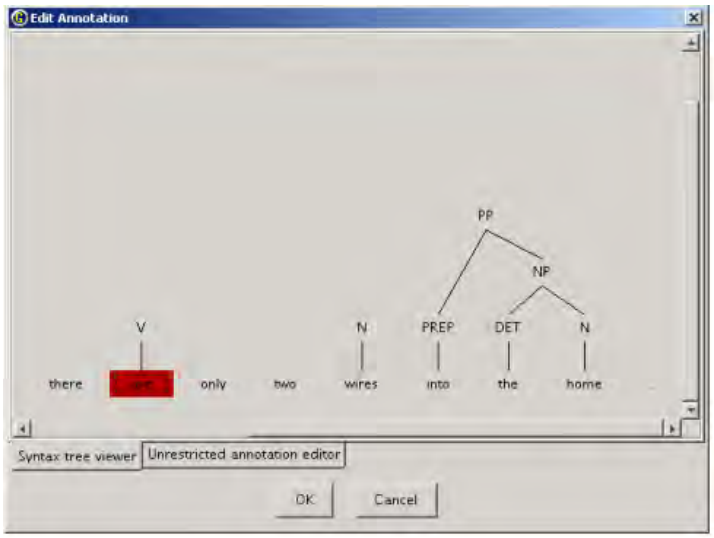

Figure 2.5: The syntax tree viewer, showing a partial syntax tree for a sentence from a telecom news text

A Nearly-New Information Extraction System (ANNIE) that is distributed with an IE system by GATE. NER is one of function in ANNIE. These recourses can be used as one unit or used as individual components along with others. ANNIE consists of the following processing components for English text [48] that we adopt for our work:

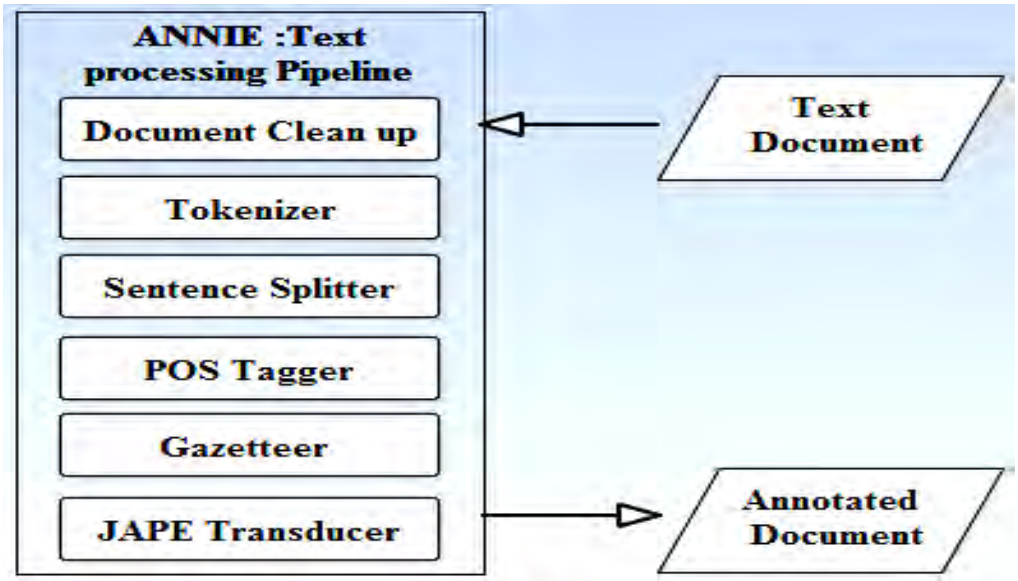

Figure 2.6: Typical GATE system components

As the text document enters the pipeline it first gets cleaned up from previous annotations; then sentences (through a Sentence Splitter resource) and words (through a Sentence Tokenizer resource) are identified; then POS tagging is performed. These constitute fundamental steps on which further processing steps can be based and which may include applying gazetteers to recognize geographic entities, proper nouns or dates. Finally, user defined elaboration processes are performed at the end of the pipeline, by means of transducers that use custom JAPE grammars that allow identifying patterns that are relevant for a certain domain and annotating them. At the end of the text processing pipeline an annotated text document is obtained and the GATE API includes Java classes that allow going through the annotations.

2.6.2. WEKA

WEKA is software which provides Machine Learning algorithms for Data Mining Applications. It is widely renowned and extensively used software in the field of Data Mining and Exploration. WEKA is ideal for our purpose as it provides many built-in classifiers for Machine Learning and

Prediction. Moreover WEKA is distributed as runnable Java Archive (jar) file, thus WEKA can easily be integrated with any Java application.

2.7. Evaluation Metrics for NER

In order to evaluate the performance of NER systems and compare it to the performance of similar systems or humans, several evaluation metrics were defined, the most popular being precision, recall and F-measure. Precision and recall are two widely used statistical classifications. In an information extraction scenario,precision is defined as the number of relevant NEs retrieved by recognizer divided by the total number of NEs retrieved by that recognizer, and Recall is defined as the number of relevant NEs retrieved by recognizer divided by the total number of existing relevant NEs (which should have been retrieved).

Where precision is the percentage of correct NEs found by the system. It can be expressed as:

$$\text{precision} = \frac{\text{Number of correct named entities found by the system}}{\text{Number of named entities found by system}}$$

and recall is the percentage of NEs existing in the corpus and which were found by the system. It can be expressed as:

$$\text{recall} = \frac{\text{Number of named entities found bythe system}}{\text{Total number of named entities}}$$

High precision may often be achieved at the expense of low recall and vice versa. A combined metric exists, F-measure, defined as:

$$\text{F-measure} = \frac{2 * p * R}{P + R}$$

Chapter Three: Related Work

In this Chapter we describe the previous research works that have been developed for Afaan Oromo and some other languages that are related and relevant to our work.

3.1. Named Entity Recognition for Afaan Oromo

Name identification has been worked on quite intensively for the past few years and has been incorporated into several products. A lot of researches in NER task have been conducted for English, European, Asian and other languages.Named Entity Recognition (AONER) system for,the third widely spoken African language, Afaan Oromo is developed by Mandefro[1]. The researcher uses ML based using Conditional Random Field (CRF)algorithm. To train the system he has developed Afaan Oromo NE corpus of more than 23,000 words out of which around 3600 are NEs. The focus was on identifying four NE types: person, location, organization and miscellaneous (date/time, monetary value and percentage).

The system function has three phases: In the first phase, called pre-processing phase, each necessary pre-processing tasks are performed on the input data, the training corpus and the plain text. Pre-processing phase consists of Parser and Tokenizer components,thus effectively utilized by the rest of the phases. Parser checks the NE tagged data whether it is legally tagged or not against the CoNLL‟s 2002 rules and standards. Tokenization used to break up a text into its constituent elements or tokens.

The second phase is called training phase, and it is used in the process of training the machine learning component. It has Begin-Inside-Outside, NE Chunker, Feature Extractor and the Model Builder components. Those components operate on data from training corpus.Begin-Inside-Outside (BIO) Encoder analyzes and identifies a token and NE tag. After identifying and tagging NE tag, the BIO encoder generates a token/tag sequence that will be supplied to both feature extractor and NE chunker.The researchers used feature extractor to identify and extract necessary features from the training data.They used CRF algorithm for feature extraction, which performs repeated extractions from the same context for different positions in the input.Building a trained model of machine learning component was the main concern of the researchers.

In the last phase, called recognition phase, NEs from tokenized plain text was recognized. This phase has Trained Model, NE Recognizer and Stored Rule components. The researchers used CRF algorithm to develop the model. Their model is trained with Afaan Oromo data; they call the trained model as Afaan Oromo CRF Model. It assists the NE Recognizer to recognize NEs from the tokenized plain texts. The last section of this phase is stored rules, and it is designed to focus on the general structure and organization of Afaan Oromo NEs. It performs the task of disambiguates, but the researchers did not cover this section in their model.

The system developed by the researchers has different components such as: parser, BIO Encoder, NE Chunker and Feature Extractor. The parser checks the correctness and validity of the corpus against CoNLL‟s standards. The BIO Encoder component identifies tokens from their tags and generates token/tag sequence.The generated token/tag sequence will be supplied to NE Chunker and Feature Extractor components. NE Chunker combines a sequence of tokens that follow each other and belong to the same NE category to create a NE chunk. Feature Extractor extracts features from the generated tokens/tag sequence. Position, word shape, POS, normalized tokens, prefix and suffix features have been extracted for each token in the training data. The outputs of the feature extractor and NE chunker are used to generate Afaan Oromo NE model trained with Afaan Oromo texts. The model generation is done by the Model Builder and it is implemented using Stochastic Gradient Descent algorithm.

Furthermore, the researchers used CRF algorithm and developedAfaan Oromo POS tagger for the generation of POS features. The researchers developedAfaan Oromo POS corpus of size around 5000 tokens to train and test the model. The efficiency of their model is 80.61% as it correctly tagged 877 tokens out of 1088 tokens. They integrated the model to AONER system to supply POS of a token to the feature extractor.

They evaluate their systems performance by using test data of 4,000 tokens. Good performance has been obtained using the majority tag concept. The evaluation of the system provides the result of 77.41% Recall, 75.80% Precision and 76.60% F1-measure.

3.2. Named Entity Recognition forArabic Language

Someresearchers have been conducted research on Arabic language NER. We present some of the successful systems that have been made on the NER of Arabic language as follows.

Benajiba et al. [2] developed machine learning based Arabic NER system on Maximum Entropy. The researchers developedresources such as Annotated Corpus for NER and a set of gazetteers.

The result of their work is acquired by assigning each word in the test set a class that was most frequently assigned to it in the training set. Then training and testing were done using the Maximum Entropy approach.

Benajiba and Rosso[4] also developed Arabic NER system using CRFs instead of Maximum Entropy. They reported that their system had significant improvement over their previouswork,Benajiba et al.[2], which is a commercial system for Named Entity Recognition. The system can recognize four types NEs: Person, Location, Organization and Miscellaneous. Arabic NER corpus consists of 316 articles which contain 150,286 tokens and 32,114 types which has a result of precision equals to 80.41% , recall equals to 67.42 and F-measure equals to 73.35.

Moreover, machine based Arabic Language NER using CRF and SVM was developed by Benajiba, Diab, and Rosso [5]. The researchers use different features to investigate the sensitivity of different NE types. They adopt Machine based, particularly SVM and CRF approaches, to build multiple classifiers for each NE type. They used ACE datasets in the evaluation process. Each NE type is sensitive to different features, and each feature plays a role in recognizing the NE in different degrees. Their result was an improvement from their previous work which Benajiba and Rosso [2] developed by using Maximum Entropy.

Elsebaiand Meziane[9] also contributed to the literature of Arabic languages. They developed rule based NER system to identify person name. Person Name Entity Recognition for Arabic Languagedeveloped by these authoris a grammar-based system built for identifying Person names in Arabic scripts with high degree of accuracy. According to them Person Name Entity Recognition for Arabic language is composed of three components: gazetteers, grammars and filtration mechanism. The gazetteer component consists of white lists of complete Person names in order to extract the matching names regardless of the grammars. Afterwards, the input text is

presented to the grammar which is in the form of regular expressions to identify the rest of Person NEs. Finally, the filtration mechanism is applied on NEs detected through certain grammatical rules in order to exclude invalid NEs. Person Name Entity Recognition for Arabic language achieved satisfactory results when applied to the ACE and Treebank Arabic Language datasets.

Omnia, Samhaa andHaggag [7] also developed system which extract person name from Arabic text. In this work, the researchers used a rule based approach combined with a statistical model called hybrid approach adopted to identify and extract name entity from Arabic texts. The system extracts three types of NEs: persons, locations and organizations. Their approach consists of two phases. In the first phase, which is also called the building of resources phase, person names are collected and clusteredand then name indicating patterns are extracted. In the second phase calledExtraction of persons‟ names phase, persons‟ names are extracted fromthe input text entriesby using name patterns and clusters which is developed in the first phase.

The researchers [7] have evaluated their system by using the precision, recall and f-score measures based on what it extracts as names from the benchmark of Arabic language NER corpus of Benajiba and Rosso [4] dataset. As mentioned earlier, Arabic NER corpus of Benajiba and Rosso [4] consists of 316 articles which contain 150,286 tokens and 32,114 types. Proper Names form 11% of the corpus.

Shamsi and Guessoum [8]are among the scholars who conducted a research on the Arabic language. They presented a statically approach that use a Hidden Markov Model (HMM) to build POS tagging of Arabic language text. They have developed their corpus of native Arabic language articles which they have manually tagged. The tagger was trained on 27594 nouns, 23554 verbs, 5722 adjectives and 5384 proper nouns.

Arabic rule based NER system using handcrafted syntactic grammars had been developed by Mesfar[10]. The rules are usually implemented in the form of regular expressions or finite state transducers. The component consists of a tokenizer, morphological analyzer and NE finder. The NE finder exploits a set of gazetteers and indicator lists to support rules construction. The system identifies NEs of types: Person, Location, Organization, Currency, and Temporal expressions. The

system utilizes the morphological information to extract unclassified proper nouns and thereby enhance the overall performance of the system, reporting f-measure of 87.3%.

3.3. NER for Indian Languages

N. Kishorjit,et.al has developed NER system for the Manipuri Language which is one of a scheduled Indian constitutionallanguage based on Bengali script[11]. The researchers used Machine learning CRF algorithm to develop their model. The model file is a readymade file by the CRF tool in the testing process. Therefore, the model file is the learnt file after the training of CRF.

Model feature selection is done through manual assumption, but implementation of a Genetic Algorithm or other technique in feature selection could be the future road map. The features in the model can be tried and implemented in other agglutinative languages specially other Indian languages such as Bengali,Iban,Tamil,Oriya and Santali.

Feature selection is an important factor in recognition of Manipuri Name Entity using CRF. The performance of themodel is evaluated by using precision, recall and F-measure. The model proved to have a Recall of 81.12%, Precision of 85.67% and F-Score of 83.33%.

The NER system for the second widely spoken Indian language, Telugu, was demonstrated by Srikanth & Murthy [12]. The system is developed in such a manner that first, they started building a Weka based Noun Tagger. Then, they developed a rule based NER system for Telugu. Their focus was mainly on identifying person, place and organization names from Telugu texts.

The system functions in two phases. In the first phase, called noun identification phase, each word in a sentence is treated as to which category it belongs (noun or not noun). This is done with the help of feature vector consisting of morphological features like length, stop words, affixes, part-of-speech, position, orthographic information and suffixes.

In the next phase, called noun classification phase, nouns which have already been identified in the first phase are checked for named entities. The classification is done using two different systems; one using heuristic based system and the other using one of the machine learning tool Weka based system. It is observed that the former system faces difficulty in differentiating ambiguities. For the

later system, it is observed that there is not much improvement in the performance of the system by including more of the neighboring words as features. The experiments conducted reveal that the performance of both systems is comparable to each other, but Weka based system outperforms the heuristic based system when gazetteer and noun tagger features are considered.

The corpus they used adds up to nearly 27.3 million words consisting of each word annotated as noun or not-noun. Trained on a manually tagged data of 13,425 words and tested on a test data set of 6,223 words, the Noun Tagger has given an F-Measure of about 92%. A manually checked NE tagged corpus of 72,157 words has been developed using the rule based tagger through bootstrapping. They have obtained overall F-measures between 80% and 97% in various experiments. The system they developeddoes not handle multi-word expressions - only individual words are recognized and partial matches are also considered as correct in their analyses.

Li and McCallum[13]also presented a Hindi NER system which was developed using CRF with feature induction. The system was required to find all appearances of three types of entities: Person, Location and Organization.

The CRF‟s tremendous freedom to include arbitrary features helped them to automatically discover relevant features by providing a large array of lexical tests though they have little knowledge of the language. While CRFs generally can use real-valued feature functions in their experiments, all features are found to be binary.

On the other hand, the ability of feature induction to automatically construct the most useful feature combinations that increase conditional likelihood let the training procedure automatically perform the feature engineering by selecting feature conjunctions that will significantly improve performance. At the beginning, they started with no features at all and choose new features iteratively. In each iteration, some set of candidates are evaluated (also using the Gaussian prior), and the best ones are added to the model. In an effort to reduce over fitting, they used a combination of a Gaussian prior and early stopping. Even though the Gaussian prior has also been thought to significantly reduce over fitting, they observed that its effectiveness for this purpose is less than widely believed. Adapting the BIO scheme, they used two kinds of labels for each entity

type: B-type for the start of an entity and I-type for the inner part. For non-entities, they used the label O.

The training set for the system consisted of 340K words. Feature induction constructed 9,697 features from an original set of 152,189 atomic features; many are position-shifted but only about 1% are useful.

Their model does not perform as well on the test set. They hypothesize that this phenomenon may be due to the significant mismatch between the training/validation data and the test data. Highest test set accuracy of their system is the F-value of 71.50%.

Deepti Chopra, et al. [14] have also developed Hybrid approach for Hindi NER system and developed a general corpus from the Hindi news papers on the web. The Named Entity tags they have used are: Name of Person, Name of Location, Time, Month, Sport, Name of Organization, Name of Vehicle, River and Quantity.

In the first phase, they have applied the Rule based heuristics or the shallow parsing technique over the Corpus. The researchers use the clue words in which the words were used to detect the Named Entities that occur just after or before the Named Entities to be identified. In the second phase, they applied Machine based Hidden Markov Model (HMM) to detect the rest of the Named Entities.

They applied each approach separately and evaluated the results. When they apply only HMM, it performs average and the accuracy of 89.7% , and when they apply only rule based Heuristics, it performs poorly and the accuracy of 47.5 but when they apply both HMM and Rule Based Heuristics it performs height and accuracy of 94.61%. This shows that when they apply hybrid approach, combined approach, it gives very good results.

3.4. NER for Slavonic

Elena Paskalevaet al. [15] have built NE recognizer system for Slavonic languages particularly Bulgaria and Romania languages. Their system decompose the input text into words and extracts each NE by referencing gazetteer lists and applying pattern matching rules. To recognise Slavonic NEs the researchers built NE recognizer using three processing resources modules: a tokeniser, a

gazetteer and a finite state transduction grammar written using JAPE which consists of two parts: corpus annotation and performance evaluation. The described modules were developed using General Architecture for Text Engineering (GATE) and the modules communicate via GATE‟s annotation API.

The corpus annotation can be done semi-automatically by running the Slavonic NE modules over the corpus and then correcting/adding new annotations manually . Performance evaluation was carried out using the evaluation tool which enables automated performance measurement and visualisation of the results, and the benchmarking tool which enables the tracking of a system's progress and regression testing. The researchers‟ main concern is recognition of three NEs: Person Names, Names of Organizations and Dates.

The researchers obtain the first and second name of person from telephone book of Sofia which consists of 330000 records containing 6500 unique first names of which 3800 are females and 2700 are males. Within these two subsets, 250 female and 230 male ambiguous names were filtered. 91% of the family names can be directly calculated through their morphological components, thus the calculation recognizes both the family name and its gender. For person first name recognition, the researchers use list of pattern matching and list of names.

Similar to proper name identification, the researchers developed three types of extraction techniques: Pattern matching in a list, morphological calculation, and specific pattern on the alphabet and punctuation level.

According to these scholars, the English date entity recognition rules are not extended to Slavonic date entity recognition because their recording format is different. For example, Slavonic does not have the am/pm distinction; they don‟t use the “/” as separator of year, month and day. There is no tradition to write the day of the week as a part of the date – that is more facultative in narrative texts.

The point is used as the standard separator of the time intervals but semicolon is not in use in Slavonic languages. For month names, Roman digits are used but abbreviations of month names are not used.

They apply JAPE rules on the large text corpora from both languages to examine the real behavior of the above mentioned Slavonic NEs. The corpora they used was equal for both Bulgarian and Romania languages, 150 000 words for both languages.

The researchers used corpus annotation and performance evaluation, which can both be done within GATE. They annotated the corpus semi-automatically by running the Slavonic NE modules over the corpus and then correcting/adding new annotations manually. Performance evaluation was carried out using the evaluation tool (Annotation Diff). The researchers‟ evaluation experiments are planned for the next stage of their work.

3.5. NER for Greek

The first research on NER of Greek was presented at the Seventh IEEE Conference on Artificial Intelligence Applications by Lisa F. Rau [40]. Rau„s paper describes a system that extracts and recognizes company names using heuristics and handcrafted rules.

A rule-based Greek NER system has also been developed by Budi and Bressan [16] in the context of the R&D project MITOS. The NER system consists of three processing stages: linguistic pre-processing, NE identification and NE classification. The linguistic preprocessing stage involves some basic tasks like tokenisation, sentence splitting, part of speech POST and stemming. Once the text has been annotated with POS tags, a stemmer is used. The aim of the stemmer is to reduce the size of the lexicon as well as the size and complexity of NER grammar. The NE identification phase involves the detection of their boundaries, i.e., the start and end of all the possible spans of tokens that are likely to belong to a NE. Classification involves three sub-stages: application of classification rules, gazetteer-based classification, and partial matching of classified NEs with unclassified ones.

3.6. NER for English

Srihari et al. [17] proposed a hybrid NER system that exploits two ML algorithms and pattern matching rules to extract NEs of several types, including Person, Location, Organization, Date, Money, Time and Percentage, from English text. The system is composed of four main modules. The first module is the pattern-matching rules which are used to extract temporal and numerical

expressions. The second module is a Maximum Entropy model that was built to utilize the gazetteers and the contextual information in order to provide a preliminary recognition of Person and Location NEs.The third module is the HMM classifier which gives the final tagging for the entities of the Person, Location and Organization types. The final module is another Maximum Entropy model produced to identify further sub-categories such as Government and Airport. The system was evaluated using the MUC-7 dataset and the results show that using the hybrid approach improves the performance in terms of accuracy. Rule-based NER systems are also developed for English language by many scholars[18, 19, 20].

3.7. NER for Korean

Seon, et al. [21] introduced a hybrid approach to tackle the problem of NER for Korean language. Two ML approaches are used by the system: Maximum Entropy and Neural Networks. The Maximum Entropy is utilized to resolve the problem of unknown words that do not exist in any of the predefined dictionaries within the system while Neural Networks use lexical information to deal with ambiguity. The pattern-selection rules are exploited to just combine adjacent words that comprise a multiword NE. The system was developed to extract NEs of the types Person, Location and Organization. In comparison, the authors utilized a pure ML approach for the actual NER while the role of the rule-based post-processing is not NER but rather setting the boundary of the identified NEs by combining them, and this might be suitable for Korean NER.

3.8. NER for Chinese

Mencius is a hybrid NER system for Chinese Language proposed by Tsai et al. [31]. The rule-based and ML-based methods are combined in Mencius in order to enhance the recognition capabilities for Person, Location, and Organization NEs. The output of the rule-based module is used as an input to the ML-based module, which is a Maximum Entropy model. The feature set consists of internal (i.e. category-dependent) features utilized to distinguish between different NE categories. The authors built their own corpus to evaluate the performance of their system. According to the empirical results, the highest performance was achieved when the hybrid system was used along with word tokenization.

The researchers proposed hybrid approach differs from the Mencius approach in two aspects – the architecture of the hybrid system and the selected feature set. Mencius is heavily dependent on the output of the rule-based module and uses it to extract the features used by the ML module, while our hybrid system does not rely only on the rule-based system‟s output in extracting the features. Moreover, the feature set of Mencius is composed of only NE category-dependent features, while the structure of feature set includes rich features consisting of language-independent and language-specific features of different types, such as morphological, contextual and word-level features.

3.9. NER for Turkish

Kucuk and Yazici [23] presented a hybrid NER system for Turkish. The main module in the system is the rule-based module which includes knowledge sources for specific domains. The original rule-based system is extended to a hybrid NER system which learns from annotated text with the purpose of improving the knowledge sources accordingly. The authors developed their own corpus for learning and evaluation purposes. The rule-based module contains dictionaries of Person, Location and Organization NEs along with pattern bases for the recognition of the aforementioned NE types as well as the numerical and temporal expressions.

The system utilizes rote learning as the ML technique. According to the empirical results, the hybrid NER system for Turkish outperforms the pure rule-based NER system in terms of accuracy. They also proposed other hybrid approach differs from their pervious approach in several aspects[61]. The flow of processing in Kucxuk and Yazıcı‟s approach [23] goes from the ML-based module to the rule-based module while in their new hybrid approach the processing of the system is converse. In the previous system, the rule-based module is heavily dependent on the output of the ML module while the modules in the new system can function separately and do not heavily rely on each other‟s output. The new hybrid system[23] is domain-independent while the Kucxuk and Yazici‟s[61] system is built with the capability of supporting four specific domains – news, financial news, historical news and child stories. The Kucxuk and Yazici‟s[23] hybrid approach is mainly a rule-based approach which employs an ML approach for the purpose of enhancing the knowledge sources within the rule-based module to support new text genres when annotated text is available while Kucxuk and Yazici‟s[61] architecture is a true hybrid in the sense that it does not favor one approach over the other.

Michal Konkol and Miloslav Konopik [24] developed Czech NER system based up on CRF-model for Czech language which outperforms all other published systems by a large margin. The researcher use four types of entities : organizations (ORG), persons (PER), locations (LOC) and miscellaneous (MISC).Performance of the system based on standard CoNLL evaluation metric has f-measure of 74.08%.

Table 3.1: Summary of related works

Paper	Algorithm	Features	Corpus	Overall Performance
Mandefro[1]	GRF	- Capitalization - Lexical - Dictionaries - External system	27,588	F1- 76.60%
Benajiba et al. [2]	MEM	- Capitalization - Lexical - Dictionaries - External system	425,000	F1-83.03%
Elena Paskaleva et al. [15]	GATE	- Gazetteer lists - Pattern matching rules - Grammar	330,000 annotation (semi-automatically)	F1- 91%
Deepti Chopra, et al. [14]	Hidden Markov Model and heuristics rule based system	- Morphological - Lexical - Orthographic - Trigger words - Gazetteer	14,012 (Manually annotated)	F1–74.61%
Tsai et al. [31]	Maximum Entropy + Rules	- Morphological - Contextual - Word-level features	Manual developed corpus	F1– 81.50%
Michal Konkol and Miloslav Konopik [24]	Weka + Rules	- Morphological - Lexical - Gazetteer	500k words of 53, 746 is annotated	F1 – 74.08%
Li and McCallum[13]	CRF + feature induction	- Capitalization - Lexicons - Conjunctions	340,000	F1- 71.50%
N. Kishorjit,et.al[11]	CRF	- Suffix - Last word - External system	45,000	F1– 83.33%
Srkanth & Murthy [12]	Weka + Manuel Rules	- Morphological - POS tag - Orthographic - Gazetteer	13,425 manualy tagd + 72,157 annotated	F1– 80% - 90%

Chapter Four: Design of Hybrid AONER

In this Chapter, we will discuss about the overall design of our proposed AONER system. First, we will illustrate the general over view of the proposed AONER system architecture, then we will describe the rule-based and machine based components along with their sub-components from the perspective of the system‟s flow of operations in the form of processes.

4.1. Architecture of AONER

Figure 5.1 illustrates the architecture of the hybrid NER system for Afaan Oromo. The system consists of two pipelined components: rule-based and ML-based Afaan Oromo NER components. The processing goes through three main phases: The rule-based component that produces NE labels based on lists of NEs, keywords and contextual rules and ML-based component post-processor which intended to make use of rule-based component‟s NE decisions as features aiming at enhancing the overall performance of the NER chore.

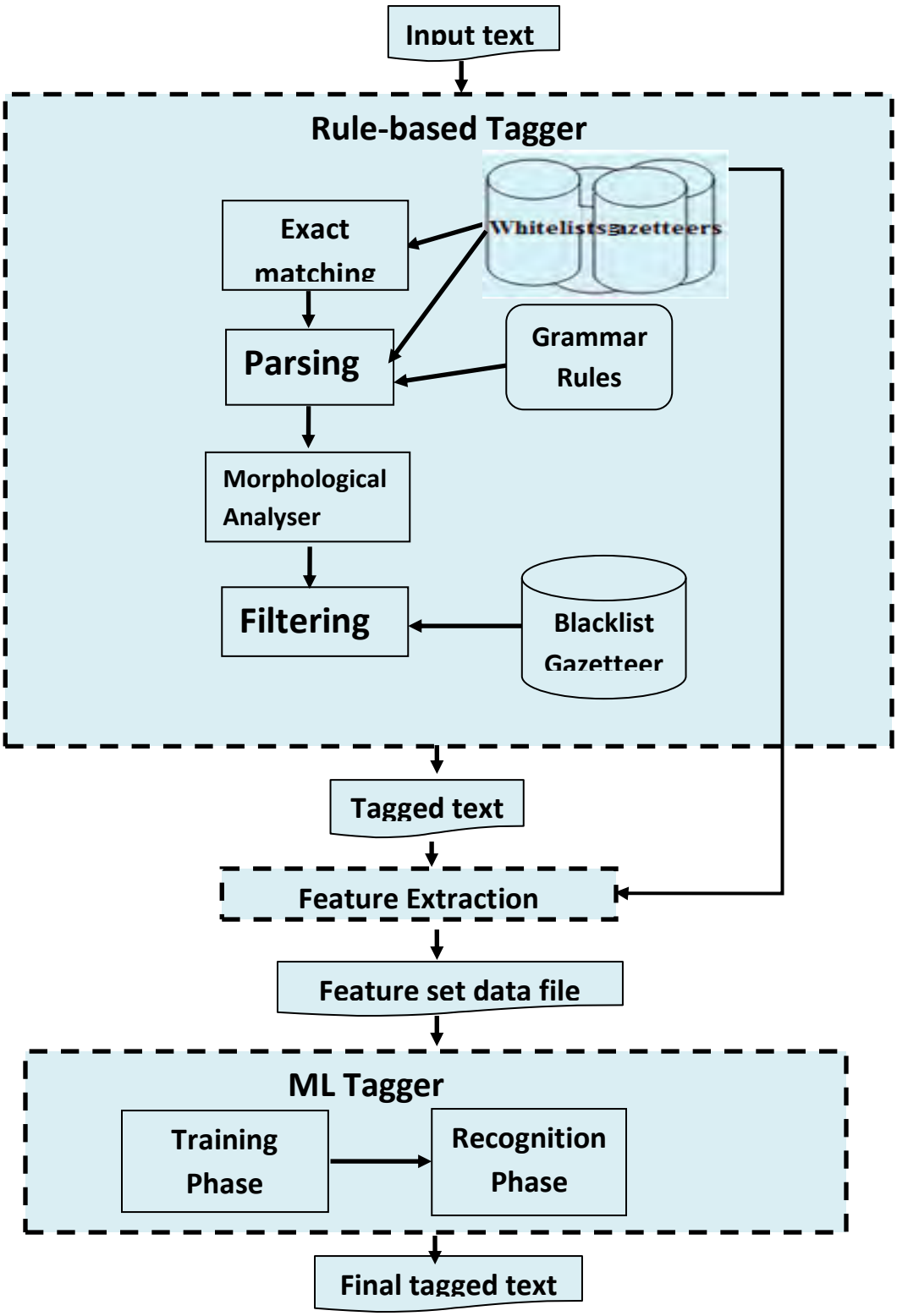

Figure: 4.1: Proposed Architecture for AONER

4.1.1. The Rule-Based Component

The rule-based component consists of three main modules: Whitelists (lists of full names), Exact matching,Grammar Rules, Parsing and a Filtering (blacklists of invalid names). We have used GATE developer tools to build and to test by using tools and processing resources like tokenizer, gazetteer and finite state transduction grammar in the rule based component. The rule-based component works as a corpus pipeline where a corpus is processed through an Afaan Oromo tokenizer, a list of gazetteers, and local grammatical Rules (implemented as finite-state transducers). The three main rule based components along with their components are discussed in the following section.

4.1.1.1. Parsing

The parser builds a semantic representation compositionally, and a „best parse" algorithm is applied to each final chart, providing a partial parse if no complete sentence span can be constructed. The parser uses a feature valued grammar. Each Category entry has the form:

Category(Feature1:Value1,...,FeatureN:ValueN)

where the number and type of features are dependent on the category type for the reason that GATE has a single model for information that describes documents, collections of documents (corpora), and annotations on documents based on attribute/value pairs.

All categories will have the features surface form and morphological root; nominal and verbal categories will also have person and number features. Verbal categories will also have tense and verb form features; and adjectival categories will have a degree feature. The noun phrase category has the same features as other nominal categories plus NE tag and named NE type.

Syntactic rules are specified in Prolog with the predicate rule (LHS and RHS), where LHS is a syntactic category and RHS is a list of syntactic categories. A rule such as BNP_HEAD => N (a basic noun phrase head is composed of a noun) is written as follows:

rule(bnp_head(sem: E^[[R,E],[number,E,N]],number:N), [n(m_root:R,number:N)]), where the feature „sem" is used to construct the semantics while the parser processes input, and E, R, and N are variables to be instantiated during parsing.

The full grammar of this distribution can be found in the prolog/grammar directory. The file load.pl specifies which grammars are used by the parser. The grammars are compiled when the system is built and the compiled version is used for parsing. Figure 4.2 shows sample parsing rules for Organization phrase parsing.

```
.cvsid_ne_rules("$Id: ne_rules.pl 7085 2005-12-05 16:32:03Z ian_roberts $").
grammar(named_entity).
rule(basic_np(edge:Edge,head:F,
        sem:E^[[name,E,F],[realisation,E,Edge],[ne_mark,E,no]]),
        [list_np(s_form:F,ne_tag:others)]).
rule(basic_np(edge:Edge,head:F2,
        sem:E^[[name,E,[F1,' ',F2]],[realisation,E,Edge],[ne_mark,E,no]]),[
organ_names_np(s_form:F1),
list_np(s_form:F2,ne_tag:others)]).
rule(basic_np(edge:Edge,head:F,
  sem:E^[[name,E,F],[organization,E],NSem,[realisation,E,Edge],[ne_tag,E,Edge]]),
  [organ_np(s_form:F,sem:E^NSem)]).
```

Figure: 4.2: Rule for organization Noun Phrase parsing

4.1.1.2. Exact Matching

The simplest pattern in JAPE is to match any single annotation of a particular annotation type. You can match only annotation types you specified in the input line at the top of the file.

4.1.1.3. Whitelists

The Whitelists are dictionaries of Named Entities or lists built for NER that are matched with target text irrespective of the rules. It is added as ANNIE (A Nearly New Information Extraction Systems) Gazetteers in GATE Developer that consists of lists like cities, organizations, days of the week, etc. It also consists of names of useful NE indicators, such as typical company designators, titles, etc. The gazetteer lists are compiled into finite state machines, which can match text tokens. The exact matches of target text with Whitelist dictionary entries are reported as Named Entities.

Tables 4.1, 4.2 and 4.3 show sample data in gazetteers for person name, organization name and location name extractor respectively.

Table 4.1: Sample Data in Gazetteers for Person Names Extractor

Name	Abbabaa Hordofaa Burqaa Gonfaa	Gammachuu Urjii Bashaadduu Caalaa	Leencoo Worqituu loomee Boonsa	Caalaa Galataa Galaanee Aadam
Person Title (pre)	obboo luba	aadde mooti	abbaa giifti	haadha goofta
	(If the next word is started with capital letter more or less the word is personal noun)			
Job Titles (pre)	doctori(Dr.) narsii abbaa seeraa	profesari(PhD.) injinari abbaa murtii	barsiisaa gaazexeessaa	barsiiftuu abbaa gada
	(If the next word is started with capital letter more or less the word is personal noun.)			
Location	Itoophiyaa Jarmanii Jaappaan Ingiliizii Jaarraa Amboo			

4.1.1.4. Grammar Rules

The grammar rules perform recognition and extraction of Afaan Oromo named entities from the input text based on derived rules. It creates annotation by describing patterns to match NEs. Grammar rules are a very important processing resource for the Afaan Oromo NER system because of its complexity.

For instance, in addition to using capitalization for proper nouns, we use NE indicators (trigger words) to formulate recognition rules, which help in identifying NEs within text. These NE indicators were obtained as a result of the deep contextual analysis of various Afaan Oromo scripts.

Afaan Oromo like many other Latin script based languages such as English has a specific signal in the orthography, namely capitalization of the initial letter. NEs are categorized under the word category of pronoun. Thus, NEs (person, location and organization) in Afaan Oromo are capitalized regardless of whether they are located at the beginning, middle or end of the sentence except date, time and monitory values.

Obsaa Chaala seena Abiishee Garbaa barreesse.

(Obsa Chala has written history of Abishe Gerba.)

Gaadisaan barata cimaadha.

(Gadisa is a strong student**.)**

In the above sentences the initial letters of words like **Obsaa, Chaala, Abiishee, Garbaa** and **Gaadisa** are capitalized because they are personal pronouns.

The rule based approach consists of a lexicon, in the form of the noun contextual clue list, together with a set of grammar rules which were responsible for recognizing and classifying NEs. As we mentioned earlier, there are clue words in Afaan Oromo to identify NEs. Some of the common clue words in Afaan Oromo are **obboo** (Mr.), **aaddee** (Mrs./Ms), **Doktor** (Dr.) for PERSON, **waldaya**, **biiroo** (office), **warshaa** (factory), **institiyuutii** (institute), **baankii** (bank), **dhaabbata** for organization, **magaalaa** (city), **waajjira**, **ganda**(kebele), **aanaa**(woreda), **godina**(zone) for location, **bara**(year), **daqiiqaa**(minute), **sokondi**(second) for date/time, „**qarshii'**, „**doolaara'** for monetary values, **qarshi**(birr), **poondi** (pound),**dolaara**(dollar) for price and **dhibbeentaa** (percent) for percentage that mostly precede NE.

Aadde Daraartuu Tulluu gara Adama deemte.

(Mrs. Deratu Tullu went to Adama).

Warshaan Daaku Adama omisha isaa dachaa sadiin dabale.

(Adama Flour Factory tripled its production).

Gamachuun Qarshii miliyoona 10 baanki Intarnaashinaal Awaashi qaba.

(Gemechu has 10 million birr in Awash International Bank.)

Affix is a language dependent feature that identifies NEs in a text. Some types of NEs often share the same affix. Afaan Oromo has some affixes that could help to identify NEs in a text. A fixed length word affix of the current and/or the surrounding word(s) can be treated as feature(s). If the length of the corresponding word is less than or equal to the original word then the feature values are not defined. The feature value is also not defined if the token itself is a punctuation symbol or contains any special symbol or digit. Another way to use the affix information is to modify the feature as binary valued. Variable length affixes of a word can be matched with predefined lists of useful affixes (e.g., "**-n**", "**-ni**", "**-i**","**-tu**", and "**-ti**"). These features are useful to handle the highly inflective Afaan Oromo language. When affixes are attached to the NEs, the word should start with capital letter as shown in the following sentence.

Tolasaan godina Balee-ti dhalate.(Tolesa born in Bale zoon.)

Gamachuu-n Qarshii miliyoona 10 Baanki Intarnaashinaal Awaashi qaba.

(Gemechu has 10 million birr in Awash International Bank.)

Moreover inflection within Afaan Oromo can be well dealt with using hand-crafted rules, which enables stripping off of the prefixes and suffixes from the stem word. Prior recognition, thus ensure the recognition of the actual NE instance alone. For each type of named entity, several rules were built and each one was applied in a particular order to ensure that the most comprehensive recognition result was achieved.

4.1.1.5. Filtering

Filtration is a mechanism that excludes the unmatched words that are recognized as NEs in the previous phase. It is performed at the last phase of NER system. The filtration is performed using blacklist dictionaries containing entries which should be rejected as Named Entities. Consider the following example:

Durataa'a boordii prezedaanti Yunivarsity Sayinsiif Teeknoloogy Adaama … [The board chair person the president of Adama Science and Technology University **…]** In this example, the words following the person indicator **Durataa'a boordii** (The board chair person) that is, **prezedaanti** (the president) is not a valid person name. The role of the blacklist, another set of rules, is rejecting such incorrect matches.

Apart from the blacklist component, certain heuristic filter rules are used for post-processing the system‟s extraction results in order to disambiguate extracted named entities. When applying a set of single-slot extraction rules to the input text i.e. sets of rules which extract particular types of named entity one after the other, one cannot exclude the possibility of identical or overlapping textual matches within the document, among different rules for different named entities. For instance, different sets of rules for extracting instances of both the named entities, person and location names may overlap or exactly match in certain text fragments, resulting in ambiguous named entities. Among these named entities, the correct choice must be made. The filter rule is an intelligent way of specifying how to get the correct choice with respect to the context in which the ambiguous situation may arise.

The following example illustrates an ambiguous situation in Afaan Oromo script:

Jaarraa Amboo afaan qorachu fedha. (Jaarra Ambo is interested to study linguistic.)

In this example the bold text fragment represents both a person name and a location. Hence when AONER is applied here, both the Person and Location Extractors will return matches as **„JaarraaAmboo'** (Jaarra Ambo). The developer can tune the system to resolve some kinds of ambiguous situations by the virtue of filter rules. One solution to disambiguate this situation is to use the following filter rule:

If a possible match 1 for a location entity intersects with a match 2 that was previously reported by the person extractor, then the match as a location name will be discarded.

Thus in case of an intersection, the match for person names is preferred over location names. The filter rules defined within the system play a significant role to handle such situations and resolve ambiguity. However, it should be built upon careful analysis of the ambiguous situations in order to get accurate results.

4.1.2. ML-Component

In this section we will discuss the Machine Learning components of AONER system. We use supervised learning for enhancing performance of the AONER system. In AONER architecture,

processes represent the evolution of the system. AONER is designed in such a manner that it first 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 (training) process and prediction process.

4.1.2.1. Training Phase

Training is only performed once to build a model in case of supervised learning. Each component of training phase for machine learning is briefly described below.

- Training phase involves presenting data to the ML algorithm from which it creates a model
- The training data (instances) have been annotated with class annotations as well as attributes
- Models are representations of decision-making processes that allow the machine learner to decide what class the instance has based on the attributes of the instance

4.1.2.2. Prediction Phase

The Prediction phase uses Model generated from training phase to predict the class of each instance in data. The Java Program can accommodate these predicted classes with data and produce final annotated text in XML format which can be utilized by range of different applications.

Chapter 5 Implementation of AONER

In this Chapter we will explain the implementation of AONER system. We used JAPE grammar for Gate Developer IDE to develop the rules for each component in the rule based part and wake for the ML part of our system.

5.1. Implementation of Rule Based Component

The rules are implemented as JAPE grammar for Gate Developer IDE. The overview of GATE Developer IDE and implementation of AONER system is described in the following sections.

5.1.1. Overview of GATE Developer IDE

As we discussed in section 2.5.1, GATE is open source text engineering software which provides a modular object-oriented framework implemented in Java to embed language processing functionality in diverse applications.

The Gate Developer is an IDE that facilitates the development of NLP systems and supports components based development. The components are referred to as CREOLE (Collection of REusable Objects for Language Engineering). It can be extended and customized for different tasks by loading plug-ins, which can in turn contain a number of resources able to hold linguistic data and to process data. GATE is distributed with an IE system called

ANNIE, which relies on finite state algorithms and the JAPE to process text corpora and performs operations such as sentence detection, tokenization, POST, chunking and parsing, named-entity detection, and pronominal co-reference [71].

5.1.2. Corpus Development

Corpora are very important Language resources and are required for linguistic studies. The corpora provide different linguistic information about the text. For our purpose Corpora that are tagged with Named Entity Information were required.

We used AONERcorp corpus prepared by Mandefro [1] for AONER; we have been used for our task beside the one we developed. The original corpus is more than 27,588 words out of which

around 3575 are NEs,from this around 23,000 words have been annotated based on CoNLL‟s 2002 BIO tagging scheme for NER. AONERcorp is not adequate for Machine Learning based system as large annotated text is required for better Machine Learning.

```
Silashii B-PER
Tasfaa I-PER
itti O
Walmorkii O
gaggeefameen O
bulchiinsi B-ORG
magaalaa I-ORG
Mattuu I-ORG
mo'ateera O
Ameerikaa B-LOC
```

Figure 5.1: ANERcorp Corpus - Sample Data

The details of AONERcorp corpus along with parsing information is described in the thesis work of Mandefro [1]. The AONERcorp is easy to parse as each line contains single word with its Entity Information. The corpus is tagged in CONLL2002 format as shown in Figure 5.1. The possible entity information attached to each tag as described in is Figure5.2 below:

```
O Words that are not named entities and referred to as 'Other'.
B-PERS Begining of Person Name
I-PERS Inside of Person Name
B-ORG Begining of Organization Name
I-ORG Inside of Organization Name
B-LOC Begining of Location Name
I-LOC Inside of Location Name
B-MISC Begining of Miscellaneous Word
I-MISC Inside of Miscellaneous Word
```

Figure 5.2: Entity information attached to each tag

5.1.3. AONER as Corpus Pipeline

In addition to the corpus developed by Mandefro[1], we use GATE Developer tool to develop AONER corpus, which is implemented as Gate Corpus Pipeline. A Corpus Pipeline is a Gate application which runs over a Corpus containing documents. Rules have been added within each phase to incorporate Whitelists into AONER rule based system within Gate Developer.
The main components of AONER Corpus Pipeline are as follows:

- Afaan Oromo Tokenizer
- Gazetteers
- Grammar Rules

6.1.3.1. Afaan Oromo Tokenizer

The Tokenizer is a processing resource used to identify tokens and their types within the target text. We adopted for our system the built-in English Language Tokenizer provided with Gate Developer LingPipe tools for our system.

The tokenizer splits text into simple tokens, such as numbers, punctuation, symbols, and words of different types. The aim is to limit the work of the tokenizer to maximize efficiency, and enable greater flexibility by placing the burden of analysis on the grammars. This means that the tokenizer does not need to be modified for different applications or text types.

As Afan Oromo writing style is similar to that of English, mainly it is the white space that demarcates the boundary of a token. With regard to NER, it is important to treat punctuation marks like periods, commas, hyphens, equal sign and double quotes as a token. This stems from the fact that they significantly affect NE tagging. Consider the following sentence:

Dr.Margaan ji'a afuriif Dhaabbata Worshaa Huccuu Aqaaqii Hojate. (Merga works at Akeki Textile Factory for four months.)

In the above Afan Oromo sentence, which has 11 tokens, the period (.) in front of the letter Dr is part of NE (with tag I-PER). But the period at the end of the sentence is not part of NE (with tag

O)[1]. However, all punctuation marks (except the apostrophe character) have been treated as tokens.. The apostrophe („) character called "**hudhaa**" is frequently seen in Afan Oromo words to specify missed consonant. If this character is treated as a token, then it will divide a single word into three different tokens with each token having no meaning. As a result, our tokenization algorithm ignores apostrophe mark and considers the word as a single token. For instance, words like **Mul'isi**, **Mul'ata**, **Bal'isa** are each treated as a single token.

6.1.3.2. Grammar Rules

Grammar rule consists of hand-crafted rules describing patterns to match, as a result annotations to be created. The rules are based on regular expressions and utilize several different dictionaries within the rules.

Patterns can be specified by describing a specific text string, or annotations previously attached to tokens (e.g. annotations created by the tokenizer, gazetteer, or document format analysis). Rule prioritization (if activated) prevents multiple assignments of annotations to the same text string.

To implement regular expression based rules in GATE Developer we use Java Annotation Patterns Engine (JAPE). JAPE provides finite state transduction over annotations based on regular expressions. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consists of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements.

We develop the rules for Person, Organization, Location and Miscellaneous Named Entities extraction by using JAPE. The Implementation of JAPE based grammar rules is explained in following sections.

Rule for Person Names recognition

The rules state that words sequence in target text is annotated as Person Name, ifthe words are found in Names Gazetteer for **MaqaaNamaa**(Person Names).

The implementation of the rule as JAPE grammar rule is straightforward in this case as shown in Figure 5.3. The left hand side of the rule is text before the arrow. The \:mq" is temporary label given to text sequence in left hand side of the rule matching the pattern. In right hand side of the rule, the \:mq" label sequence is given annotation of \Person" and \person". By applying person rule using JAPE rule as sample shown inthe Figure5.3, the outputis shown in the Figure 5.4.

```
Rule: PersonFirstRule
Priority: 30
(
({Lookup.majorType == person_full}|
{Lookup.majorType == person_first})
):mq -->
    :mq.Person = {rule= " PersonFirstRule " },:mq.Person= {rule= "
     PersonFirstRule " }
```

Figure 5.3: Example Rule for Person recognition implemented as JAPE Rule

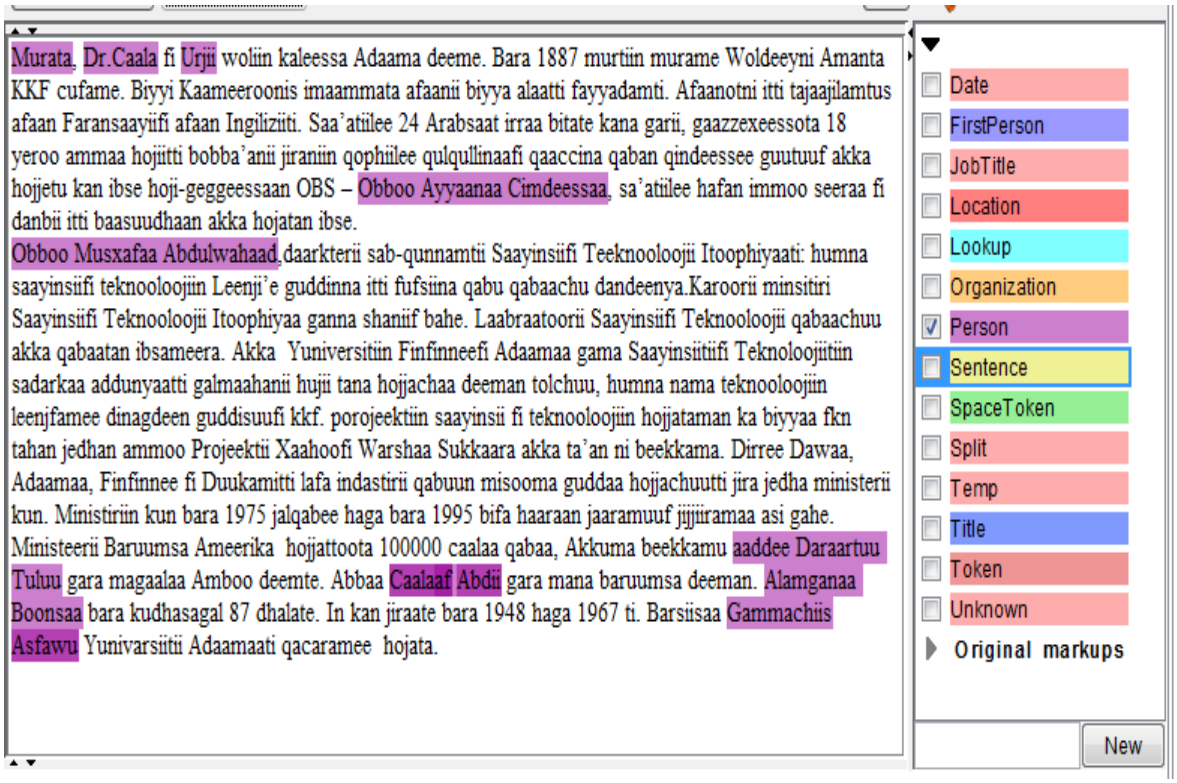

Figure 5.4: Snapshot of AONER after person name is recognized

Example rule for Person name recognition:

(surname|title)+firstName+(location)?+ (lastName)? + (number)?)

The above rule recognizes a person name composed of a first name followed by optional last name based on a preceding person indicator pattern, or the trigger words. For example, the name listed below would be recognized based on this rule.

Doctor/Dr. Caala[Doctor Chala]

Caaltuun profeesar Gammada [Chaltu ……. professor Gemeda]

Daraartuun presedenti Ameerika Obamaa woliin …… [Derartu with American President Obama]

Rule for Location Names Extractor

Apart from contextual cues, the typical Afaan Oromo naming elements were used to formulate rules such as region (**naanno**), Zoni (**godina**), Worada (**aana**) and kebele (**ganda**). There by the rules resulted in a good control over critical instances by recognizing complex entities.

Example rule for Location recognition

(**Aana**|**Godina**(Administrative division))? + (City following Indicators)? +City name+ (direction)?

The above rule recognizes a city name (existing in the dictionary of city names).

The following name would be recognized by this rule:

Godina Baalee magaala Roobe gara lixa ...[Bale zone south of Robe city …]

Some rules in Organization Name Extractor are based on clue words. The implementation of the rule as JAPE grammar is not straight forward and required JAVA code in the right hand side of the JAPE rule. The implementation of the rule is shown in Figure 6.5. By applying this rule we have the result shown in the Figure 5.6.

Example: Rule for Location Name Extractor
((**Magaalaa**)?|(**Biyyii**)? | (**Naannoo**|**Godina**)) + ((locationPostIndicatirs)?+ (City Name| Country names)

```
Rule: Location1
Priority: 65
(
{Token.category == DT})?
(
({Lookup.majorType == loc_key, Lookup.minorType == pre} )?
{Lookup.majorType == location}
(
{Lookup.majorType == loc_key, Lookup.minorType == post})?
)
:locName -->
{
gate.FeatureMap features = Factory.newFeatureMap();
gate.AnnotationSet locSet = (gate.AnnotationSet)bindings.get("locName");
```

```
gate.AnnotationSet loc = (gate.AnnotationSet)locSet.get("Lookup");
if (loc != null && loc.size()>0)
{
gate.Annotation locAnn = (gate.Annotation)loc.iterator().next();
features.put("locType", locAnn.getFeatures().get("minorType"));
}
features.put("rule", "Location1");
outputAS.add(locSet.firstNode(), locSet.lastNode(). "TempLocation",
features);
}
```

Figure 5.5: Location implementation as JAPE rule

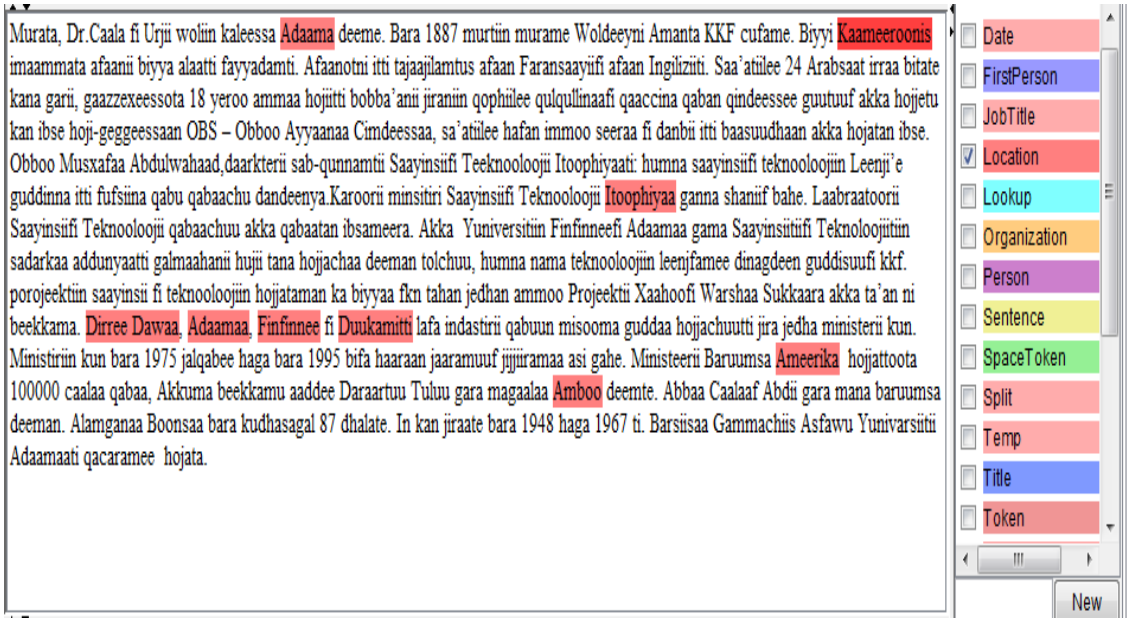

Figure 5.6: Snapshot of AONER after Location name is recognized

The rule for organization name recognition states that sequence of words in target text are annotated as organization if first word is in company following know part gazetteer, preceded by optional postfix character and followed by either an adjective of Location or Location Name. The implementation for the rule as JAPE grammar rule is given in the Figure 5.7 and the result is in the Figure 5.8.

Example: rule for organization name recognition

company_ following _known_part(**Warshaa** | **Dhaabbata**) ? + (clue words) + LocationName(country or City name)

```
Rule: org
Priority: 17
( {Lookup.majorType == "CompanyPrecedingKnownPart"}
{clueWord}
):dh-->
:dh.Organization = {rule= "dhaabbata" }
:dh.organization = {rule= "dhaabbata" }
Rule: org1
Priority: 18
({Lookup.majorType == "CompanyPrecedingKnownPart"}
{location}
):dh-->
:dh.Organization = {rule= "dhaabbata1" }
:dh.organization = {rule= "dhaabbata1" }
```

Figure 5.7: Example Organization Rule implementation as JAPE rule

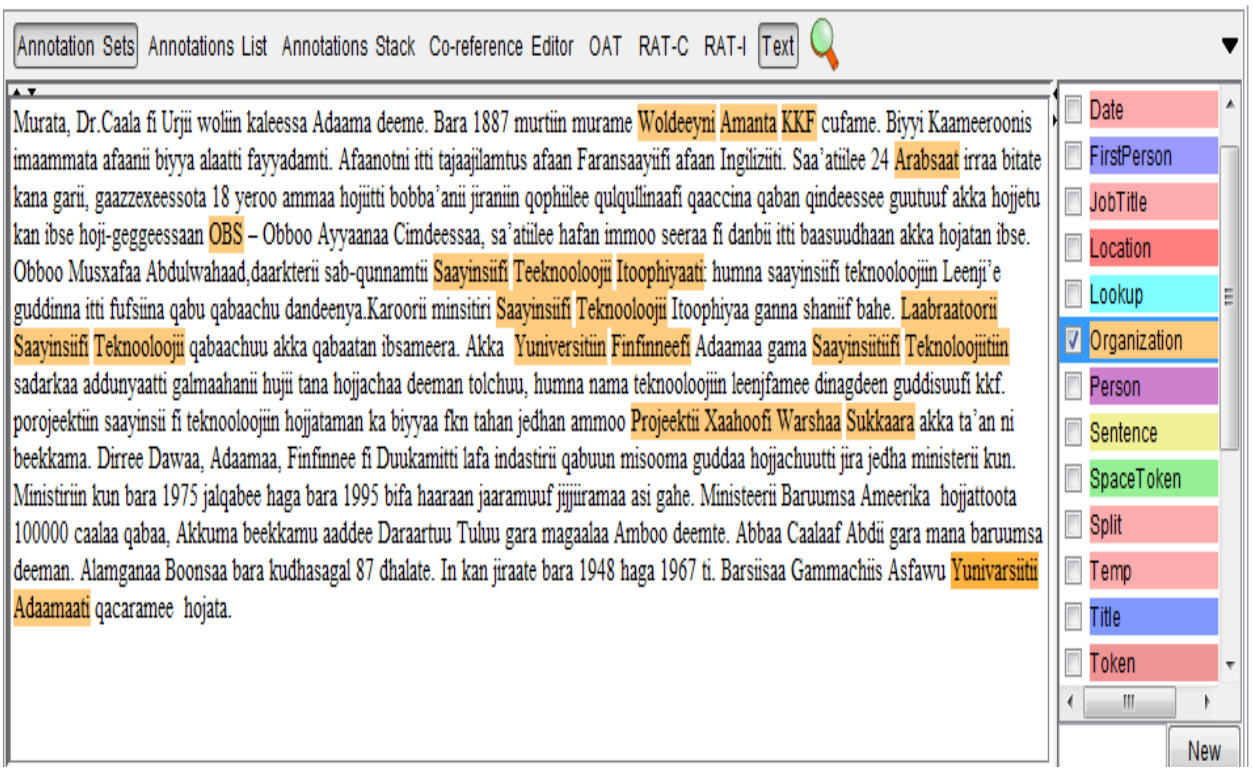

Figure 5.8: Snapshot of AONER after Location name is recognized

Incorporating Whitelist Mechanism as JAPE Grammar rules in AONER

As mentioned in Section 4.4.1 the Whitelists are added as normal Gazetteer lists in processing resources. The JAPE rule for Person Name Whitelist is illustrated in Figure5.9. The Whitelists for Location and Organization Extractors are incorporated in AONER system in the same manner. The same mechanism is applicable for Blacklist for rejecting annotations.

```
Rule: PersonRule
Priority: 30
(
    {Lookup.majorType == person_fullName}
):mq -->
  :mq.Person = {rule= " PersonRule " },:mq.Person= {rule= "
   PersonRule " }
```

Figure 5.9: Incorporating Whitelist as JAPE rules in AONER

5.2. Implementation of ML-Based Component

Results of the rule-based named entity annotation can further serve as an input for the machine learning named entity annotation. The supervised machine learning based NER consists of the following steps:

1. Preparing data for training - the data for which the values of the feature and outcome variables are recognized;

2. Using probabilistic methods-it establishes functional relationship between features and outcome (determine feature weights);

3. Using feature weights- it predicts the value of the outcome variable

5.2.1. Feature Set for Machine Learning

The ML-based component depends on two main aspects: feature engineering and selection of ML classifiers. The first aspect involves the selection and extraction of classification features. The features explored are divided into various categories: rule-based features (i.e. derived from the rule-based component‟s decisions), morphological features, POS features, Gazetteer features, contextual features, and word-level features. Exploring different types of features allow studying the effect of each feature category on the overall performance of the proposed system. The second aspect concerns the ML technique to be used in the training, testing and prediction phases. We we use Decision Trees methodes ML techniques, have been examined individually to reach a conclusion with regards to the approach to work in our hybrid system. The technique was chosen for its high performance in NER. WEKA is utilized as the environment of the ML task.

Feature is mechanisms or ways to give a clue or indication to identify NEs in a given text. Selecting the right Feature set for any Machine Learning application is a crucial task. It presents the features which are provided for each token in the input text, and improves the performance of NER system. In supervised ML, weka provides several different classifiers for identifying the NEs. Orthographic features (like capitalization, decimal, digits), affixes, left and right context (like

previous and next words), NE specific trigger words, gazetteer features, POS and morphological features etc. are generally used for NER.

We use the supervised machine learning in our hybrid AONER system in the statistical component since the supervised machine learning systems cannot be directly trained on a corpus annotated with named entities. The corpus has to be transformed into a collection of instances. During such transformation, the order of tokens and punctuation marks in the collection is not changed. We investigated a number of features and selected the following features for our AONER task.

- Rule-based feature: Our integration is done by feeding the output of the rule-based system as features to machine-learning classifiers (The Named Entity tag from rule based component is used as features in our integration to machine learning classifiers.) We call these features the rule-based features.
- Morphological features
- POS tag
- Word length flag
- Dot flag: A binary feature to indicate whether the word has adjacent dot.
- Capitalization flag: A binary feature to indicate the existence of capitalization information on the Afaan Oromo text.
- Check Gazetteers feature flags: A binary feature to represent whether the word (or left/right neighbor of targeted word) belongs to the Gazetteer set.
- Nominal flag: A binary feature to represent whether POS tag is a Noun/Proper Noun.
- Actual NE tag of the word: it is used along with other features for training the classification model. It is also used as a reference for calculating the accuracy.

5.2.2. Transformation of Corpora

In order to utilize Corpora described in previous sections, we transformed theminto XML format using JAVA code. The XML format can be used in GATE Developer. All the data of AONERcorp was transformed into single XML file. A sample transformed XML file is demonstratedin Appendix B.

5.2.3. Training

Training is only performed once to build a model in case of supervised learning.

The data used for the training of the systems was provided. The annotated data uses Shakti Standard Format (SSF). For our development we have converted the SSF format data into the IOB formatted text in which a B - XXX tag indicates the first word of an entity type XXX and I -XXX is used for subsequent words of an entity. The tag O indicates the word is outside of a NE. The training data for Hindi contains more than 5 lakh words, for Bengali about 160K words and about 93K, 64K and 36K words for Oriya, Telugu and Urdu respectively.

In time of development we have observed that the training data, provided by the organizers of the shared task, contains several types of errors in NE tagging. These errors in the training corpora affects badly to the machine learning (ML) based models. But we have not made corrections of the errors in the training corpora in time of our development. All the results shown in the paper are obtained using the provided corpora without any modification in NE annotation.

5.2.2.1. Application of Rule Based System

The corpora transformed in XML format are used as annotated corpora containing actual Named Entities. In first step of training, rule based system is applied on these annotated corpus. The output of this step is annotated text with Named Entities from rule based system. We end up with two annotated text files in XML format. One file contains actual named entities while other is annotated with named entities produced by rule based system.

5.2.2.2. Dataset Generation

The dataset is generated from the two obtained XML files in format compatible with Machine Learning tool WEKA. The two annotated files in XML format are parsed using JAXP1.3. The java program utilizes Stanford POS Tagger to tag part of speech information for each word. The

program also searches each word along with its left and right neighbors in gazetteers for Person, Location, Organization, date and time and address. Program finally produces features in comma separated values (CSV) format which can be utilized by WEKA Software.

5.2.2.3. Model Generation

Machine Learning Model is generated using WEKA by application of different classifiers. Once the model is generated, it can be saved for future use, for prediction and testing of new dataset.

5.2.4. Prediction Phase

In the Prediction phase of Machine Learning, input text is annotated by rule based; on the other hand, an annotated data can be used as an input for Prediction phase for evaluation of performance. The input for Prediction phase is usually data without known annotations. The components of Prediction phase as explained in subsequent sections are quite similar to that of Training phase.

5.2.3.1. Application of Rule Based System

In first step of training, rule based system is applied on test corpus. This process is similar to first step of training phase except that we have only one annotated file at the end of this process in case of un-annotated data. We can end up with two files in case of testing/evaluation of classifier.

5.2.3.2. Dataset Generation

The dataset generation is also similar to dataset generation for training phase. Program produce features in comma separated values (CSV) format which can be utilized by WEKA Software. The last attribute can take any dummy value for data whose actual classes/named entities are unknown.

5.2.3.3. Prediction

The Prediction phase use Model generated from training phase to predict the class of each instance in data. The Java Program can accommodate these predicted classes with data and produce final annotated text in XML format which can be utilized by range of different applications.

Chapter 6Experiments and Results

In this Chapter the performance of AONER, which was described in Chapter 6, is evaluated. To be able to obtain valid results and to interpret them properly, it is important to know how to evaluate a NER system in general, i.e. how to measure the performance of a NER application. Therefore, some general methods for evaluation are described in Section 7.1. After that, the dataset used for the experiment and results of the experiment are described and discussed in Section 7.2 and 7.3 respectively.

6.1. Evaluation Metrics

The widely used evaluation metrics i.e. Precision, Recall and F-Measure are used for system evaluation. These metrics have become standard evaluation method for Information Retrieval systems [75]. The Precision and Recall are given as:

$$\text{Precision} = \frac{\text{CorrectEntitiesIdentified}}{\text{TotalEntitiesIdentified}}$$

and recall is the percentage of NEs existing in the corpus and which were found by the system. It can be expressed as:

$$\text{Recall} = \frac{\text{CorrectEntitiesIdentified}}{\text{TotalCorrectEntities}}$$

F-Measure is a harmonic mean that weights Precision and Recall equally and isgiven as

$$\text{F-measure} = \frac{2 * p * R}{P + R}$$

6.2. Experimental Setup

In this section, we will describe the experimental setup of the system

6.2.1. Dataset Generation

For experiments the datasets are generated using Java code as illustrated in Section 6.2.2.2 using Corpora annotated with rule based. Six datasets are generated, one for each of AONER corpora data.

6.2.2. Classifier Used

WEKA provides several different classifiers that can be applied to data set; however we used the J48 as an implementation of C45 for decision trees Machine Learning.

6.2.3. Cross Validation Methodology

Cross validation is the standard way of evaluating Machine Learning systems. In order to evaluate Machine Learning system performance while avoiding the over fitting, 10 fold cross validation is used for each Machine Learning classifiers applied on every dataset. For every iteration, the system splits the dataset into ten different subsets. Nine subsets are used for training while the leftover one subset is used for prediction and evaluation. Thus, whole dataset is fairly evaluated by means of tenfold cross validation.

6.2.4. Experiments Conducted

The experiments are conducted using WEKA. The datasets are loaded in WEKA as input. Default parameters are used for each classifier used. The datasets contained instances each with features as defined in Section 4.1.1.4. The last feature\Actual" represent the target attribute for classification. Each classier is applied to dataset with three different settings of feature sets. In first setting, each

Machine Learning Classifier is applied on all the features except rule based features (feature 1). In second setting, each classifier is applied on only rule based on features. In third setting, each classifier is applied on all the features thus representing Hybrid System. The output of the

experiments include detailed accuracy measures (Precision, Recall, F-Measure, etc.), predicted class for each instance in dataset and confusion matrix etc.

6.3. Results and Discussions

The datasets generated from AONERcorp corpus are used for evaluation of systems. These corpora are described in Section 5.1.2.The results including Precision (P), recall (R) and F-Measure (F) for AONER-corpus from rule based system and Machine Learning classifiers (J48) are listed in Table 6.1. The annotations from rule based system are reported for comparison purpose in thirdcolumn. The result of rule based serves as baseline for ML. The results of Machine Learning classifiers are shown in Table6.2. In first setting each Machine Learning Classier is applied on all the features except rule based features AONER system by Mandefro[1]). In second setting each classier is applied on only rule based features. In third setting each classier is applied on all the features thus representing Hybrid System (Hybrid).

Table 6.1: Results for AONER represented in %

		AONER-Rule-based	**AONER-Hybrid**
Person	P	80.15	85.9
	R	76.22	80.78
	F	**78.14**	**83.22**
Organization	P	63.1	76.26
	R	86.50	85.99
	F	**72.97**	**80.83**
Location	P	87.12	90.6
	R	65.1	78.4
	F	**74.52**	**84.06**
Date and Time	P	86.12	86.67
	R	84.5	85.51
	F	**85.30**	**86.08**
Percentage	P	82.35	83.67

	R	79.4	81.11
	F	**80.84**	**82.37**
Monetary Value	P	80.2	83.87
	R	76.32	77.42
	F	**78.21**	**80.52**
Address	P	80.45	81.9
	R	78.56	79.3
	F	**79.49**	**80.58**
Mean	F	**78.50**	**82.52**

As evident from Table 6.1,the results from the Hybrid system outperforms as compared to individual rule based system we developed and individual Machine Learning system with only rule based features developed by Mandefro [1], in terms of F-Measure. Interestingly performance of the rule based system is very similar to performance of Machine Learning

The results from AONER systems developed by Mandefro[1], which isdeveloped by Machine Learning system using conditional random fields is listed in Table 6.2., also performed experiments on AONERcorp corpus, the results are comparable with our result. Clearly the Hybrid systems outperform for all seven Named Entity types. Moreover overall F-Measure of Hybrid system (i.e. 82.52%) outperform the overall average F-Measure of Mandefro [1] (i.e. 76.60%) for sex named entity types by 5.92 points.

Table 6.2: Results for ANERcorp by Mandefro[1] in percent

	P	R	F
AONER	77.84	75.59	76.70

Chapter 7: Conclusion and Recommendation

7.1. Conclusion

A hybrid approach is the most recent approach which integratesrule-based with ML approaches. The integration is more intuitive and linguistically motivated as it conducts an Afaan Oromo NER pipeline that combines rule-based features with other features used in machine learning. The proposedhybrid system has achieved an overall improvement of the AONER performance. It iscapable of recognizing person name, organization name, location name, date time, currency and address name.

Anumber of extensive experiments have been conducted on three different dimensions including thenamed entity types, the feature set (divided into groups) and the ML technique to evaluate theperformance of our Afaan Oromo NER system when applied on different datasets. The experimentalresults show that the hybrid approach outperforms the pure Rule-based approach and the pureML-based approach. Our hybrid NER system for Afaan Oromooutperforms the state-of-the-art of theAfaan Oromo NER in terms of f-measure when applied to AONERcorp dataset with f-measure of 83.22% for Person named entities, f-measure of 84.06% for Location named entities, f-measure of 80.83% for Organization named entities, f-measure of 86.06% for Date and Time named entities, f-measure of 83.67% for Percentage named entities and f-measure of 80.58% for Address named entities.

7.2. Future work

The developed AONER has portions that require further improvements that we want to recommend them as future work. Hence, the following recommendations are made for further research and improvement:

In future work, we recommend linguists to enhance the gazetteers and explore the possibility of improving the system with adding more lists. There is also a space for improving the grammatical rules implemented within the rule-based component through analyzing the hybrid system‟s output in away to automate the enhancement process.

The system developed in this research work is just a prototype. Any interested person can do a project to develop a full-fledged Afaan Oromo NER that can be easily integrated into different Afaan Oromo NLP works like;

- Grammar checkers
- Search engines
- Question Answering
- Text to speech synthesis
- Machine translation
- Dictionaries and the like

References

[1] Mandefro Legesse, Named Entity Recognition for Afaan Oromo, Master‟s Thesis, School of Graduate studies, Addis Ababa University, 2012.

[2] Benajiba.Y., P.Rosso, J.M.Benedi Ruiz, An Arabic Named Entity Recognition System Based on Maximum Entropy, Springer Heidelberg, 2007.

[3] Assefa W/mariam, Developing Morphological Analysis for Afaan Oromo Text, Master‟s Thesis, School of Graduate studies, Addis Ababa University, 2005.

[4] Benajiba Y. and Rosso P., Arabic named entity recognition using conditional random fields,2008.

[5] Benajiba, Y., Diab, M. and Rosso, P., Arabic named entity recognition using optimized feature sets, in Proceedings of the Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, 2008.

[6] Chinchor, N., Overview of MUC-7/MET-2, in Proceedings of the 7th Message Understanding Conference, 1998.

[7] Omnia Zayed, Samhaa El-Beltagy and Osama Huggag., A Novel Approach to Detecting Arabic Person‟s Names using Limited Resources, Center of Information Science, Egypt, 2013.

[8] Al Shamsi Fatma, and Ahmed Guessoum, A hidden Markov model-based POS tagger for Arabic, in Proceeding of the 8th International Conference on the Statistical Analysis of Textual Data, France, 2006.

[9] Elsebai, A., Meziane, F., Extracting person names from Arabic newspapers,in Proceedings of the International Conference on Innovations in Information Technology, pp:87–89.UAE, 2011.

[10] S. Mesfar, Named entity recognition for arabic using syntactic grammars, in Lecture Notes in Computer Science, Berlin, Heidelberg, pp: 305-316,2007.

[11] N. Kishorjit and S. Bandyopadhyay, Identification of MWEs Using Weka in Manipuri and Improvement Using Reduplicated MWEs, in the Proceedings of 8th International Conference on Natural Language, IIT Kharagpur, India, 2, pp 51-57,2010.

[12] Srikanth, P and Kavi N. Murthy, Named Entity Recognition for Telugu, in Proceedings of the International Journal of Coptitional Nanguage Processing, 8,pp:41–50, Hyderabad, India, 2008.

[13] W. Li and A. McCallum, Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction (Short Paper), 2003.

[14] Deepti Chopra, NusratJahan, and SudhaMorwal, Hindi named entity recognition by aggregating rule based heuristics and hidden markov model, International Journal of Information, 2(6), 2012.

[15] Elena Paskaleva, Galia Angelova, Milena Yankova, Kalina Bontcheva, Hamish Cunningham and Yorick Wilks, Slavonic Named Entites in GATE,2010.

[16] I. Budi, S. Bressan, Association Rules Mining for Name Entity Recognition, Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003.

[17] Srihari, Rohini, Cheng Niu, and Wei Li., A hybrid approach for named entity and sub-type tagging, Proceedings of the sixth conference on Applied natural language processing,.Association for Computational Linguistics, 2000.

[18] Aone, C., Halverson, L., Hampton, T. and Ramos-Santacruz, Description of the Information Extracttion system used for MUC-7,2(1), Fairfax, Virginia, 1998.

[19] Mikheev, A., Grover, C., Moens, M., Description of the LTG system used for MUC-7. in MUC-7, Fairfax, Virginia, 1998.

[20] Mikheev, A., Grover, C., Moens, M., Named Entity Recognition without Gazeteers, in Proceedings of EACL,8(1), Bergen, Norway,1999.

[21] Seon, Choong-Nyoung, Youngjoong Ko, Jeong-Seok Kim, and Jungyun Seo., Named Entity Recognition using Machine Learning Methods and Pattern-Selection Rules, 2001.

[22] Kaiser, K., & Miksch, S, Information Extraction, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, 2005.

[23] Kucuk, Dilek, and Adnan Yazici, A hybrid named entity recognizer for Turkish with applications to different text genres, in Computer and Information Sciences,2(3):113-116, Netherlands, 2010.

[24] Michal Konkol and Miloslav Konopik, CRF-based Czech Named Entity ecognizer and Consolidation of Czech NER Research, Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia,2010.

[25] Bikel, Daniel M., Miller, S., Schwartz, R. and Weischedel, R. 1997, A High-Performance Learning Name-finder,2008.

[26] Tesfaye Semela, Intergroup relations among the Ethiopian youth effects of ethnicity, language and religious backgrounds, in Journal of Development Studies, Lose Angles, Sage Publications, 28 (3): 323-354, 2012.

[27] Jay Friedenberg and Gordon Silverman, An Introduction to the Study of Mind, Cognitive Science, SAGE, 2006.

[28] Jordi Turmo and Alicia Ageno, Adaptive Information Extraction, Catalanian Research Center, Universitat Politecnica de Catalunya, Spain,2007.

[29] Liddy, E.D., Natural Language Processing, in Encyclopedia of Library and Information Science, Marcel Decker, NY, 2(1), 2001.

[30] *Tabor Wami's,* new book titled, Yewugena Dirsetoch ena Yetarik Ewunetawoch,2004.

http://www.omniglot.com/writing/oromo.htm last accessed Friday, october 31,2014.

[31] Tsai, Tzong-Han, Shih-Hung Wu, Cheng-Wei Lee, Cheng-Wei Shih, and Wen-Lian Hsu, A Chinese named entity recognizer using the maximum entropy-based hybrid model, International Journal of Computational Linguistics and Chinese Language Processing, 9(1), 2004.

[32] Ekbal, A. and Bandyopadhyay, S., Named Entity Recognition using Support Vector Machine , 2010.

[33] Louis, A., De Waal, A., and Venter, C., Named Entity Recognition in a South African Context, Proceedings of the annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries, Somerset West, South Africa, pp: 170 – 179, 2006.

[34] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham and Yorick Wilks, Named Entity Recognition from Diverse Text Types, Departement. of Computer Science, University of Shefield, UK, 2001.

[35] Grishman, Ralph and Sundheim, B., Message Understanding Conference – 6, A Brief History, in Proceedings International Conference on Computational Linguistics,1996.

[36] Chinchor, N., MUC-7, Named Entity Task Definition Dry Run Version, Proceedings of the Seventh Message Understanding Conference, Fairfax, Virginia, 1997.

[37] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R.,The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation, In Proceedings of the Conference on Language Resources and Evaluation LREC, Lisbon, Portugal, 4, pp:837-840,2004.

[38] Hirschman L., and Chinchor, N, Named entity task definition, 1998.

[39] Tjong Kim Sang, Erik. F., Language-Independent Named Entity Recognition, 2002.

[40] Rau, Lisa F., Extracting Company Names from Text, Proceding Conference on Artificial Intelligence Applications of IEEE, 1991.

[41] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R., Ontonotes, the 90% solution, in Proceedings of the Human Language Technology Conference of the NAACL, Association for Computational Linguistics, 2006.

[42] Weischedel, R., and Brunstein, A., Bbn pronoun coreference and entity type corpus, Linguistic Data Consortium, Philadelphia, 2005.

[43] Hoffmann, R., Zhang, C., and Weld, D., 2010 Learning 5000 relational extractors, in Association for Computational Linguistics, 2010.

[44] Alfonseca, Enrique, and Suresh Manandhar, An unsupervised method for general named entity recognition and automated concept discovery, Proceedings of the 1st international conference on general WordNet, Mysore, India, 2002.

[45] Sekine, Satoshi, and Nobata, C., Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy, 2004.

[46] Appelt, D. E., & Israel, D. J., 1999 Introduction to information extraction, Proceedings of the International Joint Conference on Artificial Intelligence,1999.

[47] Razvan, C. Bunescu., Learning for Informat ion Extraction, From Named Entity Recognition and Disambiguation to Relation Extraction, The University of Texas at Austin, Texas, 2007.

[48] H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan, Gate, A framework and graphical development environment for robust natural language processing tools and applications,in Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics' 2002.

[49] Zhou, G. and Su, J., Named Entity Recognition using an HMM-based Chunk Tagger, in Proceedings of Association for Computational Linguistics, Philadelphia, pp: 473–480, 2002.

[50] Brew, C., Statistical methods of Language processing, 2006.

[51] Sarawagi, Sunita, and William W. Cohen., Semi-markov conditional random fields for information extraction, Advances in Neural Information Processing Systems, 2004.

[52] Sekine, S. and Ranchhod, E., Named Entities Recognition, classification and use, Special issue of Lingvisticæ Investigationes, 30(1): 3-26,2007.

[53] Yu, Shihong, Bai S., Wu, P., Description of the Kent Ridge Digital Labs System Used or MUC-7, in Proceedings Message Understanding Conference, 1998.

[54] Bick, Eckhard, A Named Entity Recognizer for Danish, in Proceedings Conference on Language Resources and Evaluation, 2004.

[55] Ruch Patrick, Robert Baud, and Antoine Geissbühler, Using lexical disambiguation and named-entity recognition to improve spelling correction in the electronic patient record, Artificial intelligence in medicine, 29(1):169-184,2003.

[56] S. Coates Stephens, The analysis and acquisition of Proper names for the understanding of free text, in Computers and the Humanities, Kluwer, Hingham, 26(5–6):441–456, 1993.

[57] Jansche Martin, Named entity extraction with conditional markov models and classifiers, in the proceedings of Association for Computational Linguistics on Natural language learning, ,20(6), 2002.

[58] Collins Michael, Ranking Algorithms for Named–Entity Extraction, Boosting and the Voted Perceptron, in proceedings Association for Computational Linguistics, 2002.

[59] Raghavan, Hema, James Allan, and Andrew McCallum, An exploration of entity models, collective classification and relation description, 2004.

[60] N.F. Noy, M. Sintek, S. Decker, M. Crubzy, R.W. Fergerson, and M.A. Musen, Creating Semantic Web Contents with Prot´eg´e-2000, IEEE Intelligent Systems, 16(2):60–71, 2001.

[61] Küçük, Dilek, and Adnan Yazıcı., A hybrid named entity recognizer for Turkish, Expert Systems with Applications. **39**(3): 2733-2742, 2012.

[62] Baye Yimam, Yamarinna Sáwasáw (Amharic Grammar), Addis Ababa.EMPDA, 1987 (E.C.).

[63] Mohammed Hassen, 'A Brief Glance at the History of the Growth of Written Oromo Literature' in Cushitic and Omotic Languages 3rd, International Symposium, Berlin, 1996.

[64] Gragg, G., Oromo Dictionary, The African Studies Center, Michigan State University, 1982.

[65] Tilahun Gamta, The Oromo language and the latin alphabet, Journal of Oromo Studies, 1992.http://www.africa.upenn.edu/Hornet/Afaan_Oromo_19777.html last visited on Friday, October 31, 2014.

http://ethnomed.org/culture/oromo/oromo-alphabets-and-sounds-sagaleewani-fi-loqoda/, last visited on Friday, October 31, 2014.

[66] Yenewodim Biadgie, Application of Multilayer Neural Network for Tagging Parts of Speech for Amharic Language, Masters Thesis, School of Graduate studies, Addis Ababa niversity, 2006.

[67] Claude Hagège, Adpositions, Function Marking in Human Languages , Oxford Studies in Typology and Linguistic Theory, ,2010.

[68] Zwart, Jan-Wouter, A note on functional adpositions, Organizing Grammar and Linguistic Studies, Berlin, 2005.

[69] Getachew Mamo, Automatic Part Of Speech Tagging for Afaan Oromo Language, Masters Thesis, School of Graduate studies, Addis Ababa University, 2009.

[70] Tilahun Gamta, The Oromo language and the latin alphabet, Journal of Oromo Studies, 1992.

[71] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks, Named Entity Recognition from Diverse Text Types, in Recent Advances in Natural Language Processing Conference, Tzigov Chark, Bulgaria, 2001.

[72] Cooper, R.L., Language planning and social change, Cambridge University Press, Cambridge, 1989.

[73] Jeylan W.H., The Politics of Language, Power and Pedagogy in Ethiopia, Addressing the Past and Present Conditions of the Oromo Language, Australian Journal of Linguistics, **28**(1):31-57, 2008.

[74] Kebede Hordofa, Towards the Genetic Classification of the Afaan Oromoo Dialects, The University of Oslo, Department of Linguistics and Scandinavian Studies, 2009.

[75] De Sitter A., Calders, T. and Daelemans, W., A formal framework for evaluation of information extraction, 2004.

APPENDICES

Appendix A: Sample Data in Afaan Oromo Gazetteers

A.1 Sample data Gazetteers for Person Names Extractor

Name	Honorifics (post)	Person Title (pre)	Job Titles (pre)
Abbabaa	Arjaa	Mrs	Doctora
Gammachuu	Gowwaa	Qeesii	Profesara
Leencoo	Qaroo/Qaruutee	Daaqonii	Barsiisaa
Caalaa	Doyna/Donna	Sheekii	Barsiiftuu
Hordofaa	Goota	Obbo	Narsii
Urjii	Sodaattuu	Aadde	Injinara
Worqituu	Abshaala	Abbaa	Daldalaa
Galataa	Halbaadhessa	Haadha	Gaazexeessaa
Burqaa	Dabeessa/lunna	Eessumoo	Abbaa seeraa
Bashaadduu	Hamaa	Adaadoo	Abbaa murtii

A2. Sample Data in Gazetteers for Location Names Extractor

Country Names	City Names	Capital Names
Itoophiyaa	***Adaamaa***	***Finfinnee***
Ameerikaa	Alamata	***Landan***
Jarmanii	Amboo	***Waashingitan***
Inglizii	Arbaa Minchi	**Abu Dhaabii**
Jaappan	Asoosaa	**Abujaa**
Chaayinaa	***Asallaa***	**Akraa**
Meeksikoo	Asasaa	**Aljersi**
Kanaadaa	Asaayitaa	**Amaan**

Appendix B: Sample data annotated by GATE

This data is a tagged named entity by GATE and has an xml format which is opend by JAXP

<?xml version="1.0" encoding="WINDOWS-1252"?><GateDocument version="2"><!-- The document's features--><GateDocumentFeatures><Feature><Name className="java.lang.String">AUTHOR</Name><Value className="java.lang.String">HP</Value></Feature><Feature><Name className="java.lang.String">TIKA_CREATOR</Name><Value className="java.lang.String">HP</Value></Feature><Feature><Name className="java.lang.String">gate.SourceURL</Name><Value className="java.lang.String">file:/D:/nlp/named/nlp/Oro%20NER/Afaan%20Oromo%20gazetteer/Murata1.docx</Value></Feature><Feature><Name className="java.lang.String">MimeType</Name><Value className="java.lang.String">application/vnd.openxmlformats-officedocument.wordprocessingml.document</Value></Feature><Feature><Name className="java.lang.String">CREATOR</Name><Value className="java.lang.String">HP</Value></Feature><Feature><Name className="java.lang.String">TIKA_AUTHOR</Name><Value className="java.lang.String">HP</Value></Feature></GateDocumentFeatures><!-- The document content area with serialized nodes --><TextWithNodes><Node id="0"/>Murataa<Node id="7"/>,<Node id="8"/><Node id="9"/>Dr<Node id="11"/>.<Node id="12"/>Caalaa<Node id="18"/><Node id="19"/>fi<Node id="21"/><Node id="22"/>Urjii<Node id="27"/><Node id="28"/>woliin<Node id="34"/><Node id="35"/>kaleessa<Node id="43"/><Node id="44"/>Adaama<Node id="50"/><Node id="51"/>deeme<Node id="56"/>.<Node id="57"/><Node id="58"/>Bara<Node id="62"/><Node id="63"/>1887<Node id="67"/><Node id="68"/>murtiin<Node id="75"/><Node id="76"/>murame<Node id="82"/><Node id="83"/>Woldeeyni<Node id="92"/><Node id="93"/>Amanta<Node id="99"/><Node id="100"/>KKF<Node id="103"/><Node id="104"/>cufame<Node id="110"/>.<Node id="111"/><Node id="112"/>Biyyi<Node id="117"/><Node id="118"/>Kaameeroonis<Node id="130"/><Node id="131"/>imaammata<Node id="140"/><Node id="141"/>afaanii<Node id="148"/><Node id="149"/>biyya<Node id="154"/><Node id="155"/>alaatti<Node id="162"/><Node id="163"/>fayyadamti<Node id="173"/>.<Node id="174"/><Node id="175"/>Afaanotni<Node id="184"/><Node id="185"/>itti<Node id="189"/><Node id="190"/>tajaajilamtus<Node id="203"/><Node id="204"/>afaan<Node id="209"/><Node id="210"/>Faransaayiifi<Node id="223"/><Node id="224"/>afaan<Node id="229"/><Node id="230"/>Ingiliziiti<Node id="241"/>.<Node id="242"/><Node id="243"/>Sa<Node id="245"/>‘<Node id="246"/>atii<Node id="250"/><Node id="251"/>24<Node id="253"/><Node id="254"/>Arabsaat<Node id="262"/><Node id="263"/>irraa<Node id="268"/><Node id="269"/>bitate<Node id="275"/><Node id="276"/>kana<Node

id="280"/><Node id="281"/>garii<Node id="286"/>,<Node id="287"/><Node
id="288"/>gaazexeessitoota<Node id="304"/><Node id="305"/>18<Node id="307"/><Node
id="308"/>yeroo<Node id="313"/><Node id="314"/>ammaa<Node id="319"/><Node
id="320"/>hojiitti<Node id="328"/><Node id="329"/>bobba<Node id="334"/>‘<Node
id="335"/>anii<Node id="339"/><Node id="340"/>jiraniin<Node id="348"/><Node
id="349"/>qophiilee<Node id="358"/><Node id="359"/>qulqullinaafi<Node id="372"/><Node
id="373"/>qaaccina<Node id="381"/><Node id="382"/>qaban<Node id="387"/><Node
id="388"/>qindeessee<Node id="398"/><Node id="399"/>guutuuf<Node id="406"/><Node
id="407"/>akka<Node id="411"/><Node id="412"/>hojjetu<Node id="419"/><Node
id="420"/>kan<Node id="423"/><Node id="424"/>ibse<Node id="428"/><Node id="429"/>hoji-
gaggeessaan<Node id="445"/><Node id="446"/>OBS<Node id="449"/><Node id="450"/>–
<Node id="451"/><Node id="452"/>obboo<Node id="457"/><Node id="458"/>Ayyaanaa<Node
id="466"/><Node id="467"/>Cimdeessaa<Node id="477"/>,<Node id="478"/><Node
id="479"/>sa<Node id="481"/>‘<Node id="482"/>atiilee<Node id="489"/><Node
id="490"/>hafan<Node id="495"/><Node id="496"/>immoo<Node id="501"/><Node
id="502"/>seeraa<Node id="508"/><Node id="509"/>fi<Node id="511"/><Node
id="512"/>danbii<Node id="518"/><Node id="519"/>itti<Node id="523"/><Node
id="524"/>baasuudhaan<Node id="535"/><Node id="536"/>akka<Node id="540"/><Node
id="541"/>hojatan<Node id="548"/><Node id="549"/>ibse<Node id="553"/>.<Node
id="554"/><Node id="555"/><Node id="556"/>Obboo<Node id="561"/><Node
id="562"/>Musxafaa<Node id="570"/><Node id="571"/>Abdulwahaad<Node id="582"/>,<Node
id="583"/>daarkterii<Node id="593"/><Node id="594"/>sab-qunnamtii<Node id="607"/><Node
id="608"/>Saayinsiifi<Node id="619"/><Node id="620"/>Teeknooloojii<Node id="633"/><Node
id="634"/>Itoophiyaati<Node id="646"/>:<Node id="647"/><Node id="648"/>humna<Node
id="653"/><Node id="654"/>saayinsiifi<Node id="665"/><Node id="666"/>teknooloojiin<Node
id="679"/><Node id="680"/>leenji<Node id="686"/>‘<Node id="687"/>e<Node
id="688"/><Node id="689"/>guddinna<Node id="697"/><Node id="698"/>itti<Node
id="702"/><Node id="703"/>fufsiina<Node id="711"/><Node id="712"/>qabu<Node
id="716"/><Node id="717"/>qabaachu<Node id="725"/><Node id="726"/>dandeenya<Node
id="735"/>.<Node id="736"/>Karoorii<Node id="744"/><Node id="745"/>minsitiri<Node
id="754"/><Node id="755"/>Saayinsiifi<Node id="766"/><Node id="767"/>Teknooloojii<Node
id="779"/><Node id="780"/>Itoophiyaa<Node id="790"/><Node id="791"/>ganna<Node
id="796"/><Node id="797"/>shaniif<Node id="804"/><Node id="805"/>bahe<Node
id="809"/>.<Node id="810"/><Node id="811"/>Laabraatoorii<Node id="824"/><Node
id="825"/>Saayinsiifi<Node id="836"/><Node id="837"/>Teknooloojii<Node id="849"/><Node
id="850"/>qabaachuu<Node id="859"/><Node id="860"/>akka<Node id="864"/><Node
id="865"/>qabaatan<Node id="873"/><Node id="874"/>ibsameera<Node id="883"/>.<Node
id="884"/><Node id="885"/>Akka<Node id="889"/><Node id="890"/><Node
id="891"/>Yuniversitiin<Node id="904"/><Node id="905"/>Finfinneefi<Node id="916"/><Node
id="917"/>Adaamaa<Node id="924"/><Node id="925"/>gama<Node id="929"/><Node
id="930"/>Saayinsiitiifi<Node id="944"/><Node id="945"/>Teknoloojiitiin<Node
id="960"/><Node id="961"/>sadarkaa<Node id="969"/><Node id="970"/>addunyaatti<Node
id="981"/><Node id="982"/>galmaahanii<Node id="993"/><Node id="994"/>hujii<Node
id="999"/><Node id="1000"/>tana<Node id="1004"/><Node id="1005"/>hojjachaa<Node
id="1014"/><Node id="1015"/>deeman<Node id="1021"/><Node id="1022"/>tolchuu<Node
id="1029"/>,<Node id="1030"/><Node id="1031"/>humna<Node id="1036"/><Node

id="1037"/>nama<Node id="1041"/><Node id="1042"/>teknooloojiin<Node id="1055"/><Node id="1056"/>leenjfamee<Node id="1066"/><Node id="1067"/>dinagdeen<Node id="1076"/><Node id="1077"/>guddisuufi<Node id="1087"/><Node id="1088"/>kkf<Node id="1091"/>.<Node id="1092"/><Node id="1093"/>porojeektiin<Node id="1105"/><Node id="1106"/>saayinsii<Node id="1115"/><Node id="1116"/>fi<Node id="1118"/><Node id="1119"/>teknooloojiin<Node id="1132"/><Node id="1133"/>hojjataman<Node id="1143"/><Node id="1144"/>ka<Node id="1146"/><Node id="1147"/>biyyaa<Node id="1153"/><Node id="1154"/>fkn<Node id="1157"/><Node id="1158"/>tahan<Node id="1163"/><Node id="1164"/>jedhan<Node id="1170"/><Node id="1171"/>ammoo<Node id="1176"/><Node id="1177"/>Projeektii<Node id="1187"/><Node id="1188"/>Xaahoofi<Node id="1196"/><Node id="1197"/>Warshaa<Node id="1204"/><Node id="1205"/>Sukkaara<Node id="1213"/><Node id="1214"/>akka<Node id="1218"/><Node id="1219"/>ta<Node id="1221"/>'<Node id="1222"/>an<Node id="1224"/><Node id="1225"/>ni<Node id="1227"/><Node id="1228"/>beekkama<Node id="1236"/>.<Node id="1237"/><Node id="1238"/>Dirree<Node id="1244"/><Node id="1245"/>Dawaa<Node id="1250"/>,<Node id="1251"/><Node id="1252"/>Adaamaa<Node id="1259"/>,<Node id="1260"/><Node id="1261"/>Finfinnee<Node id="1270"/><Node id="1271"/>fi<Node id="1273"/><Node id="1274"/>Duukamitti<Node id="1284"/><Node id="1285"/>lafa<Node id="1289"/><Node id="1290"/>indastirii<Node id="1300"/><Node id="1301"/>qabuun<Node id="1307"/><Node id="1308"/>misooma<Node id="1315"/><Node id="1316"/>guddaa<Node id="1322"/><Node id="1323"/>hojjachuutti<Node id="1335"/><Node id="1336"/>jira<Node id="1340"/><Node id="1341"/>jedha<Node id="1346"/><Node id="1347"/>ministerii<Node id="1357"/><Node id="1358"/>kun<Node id="1361"/>.<Node id="1362"/><Node id="1363"/>Ministiriin<Node id="1374"/><Node id="1375"/>kun<Node id="1378"/><Node id="1379"/>bara<Node id="1383"/><Node id="1384"/>1975<Node id="1388"/><Node id="1389"/>jalqabee<Node id="1397"/><Node id="1398"/>haga<Node id="1402"/><Node id="1403"/>bara<Node id="1407"/><Node id="1408"/>1995<Node id="1412"/><Node id="1413"/>bifa<Node id="1417"/><Node id="1418"/>haaraan<Node id="1425"/><Node id="1426"/>jaaramuuf<Node id="1435"/><Node id="1436"/>jijjiiramaa<Node id="1447"/><Node id="1448"/>asi<Node id="1451"/><Node id="1452"/>gahe<Node id="1456"/>.<Node id="1457"/><Node id="1458"/>Ministeerii<Node id="1469"/><Node id="1470"/>Baruumsa<Node id="1478"/><Node id="1479"/>Ameerika<Node id="1487"/><Node id="1488"/><Node id="1489"/>hojjattoota<Node id="1500"/><Node id="1501"/>100000<Node id="1507"/><Node id="1508"/>caalaa<Node id="1514"/><Node id="1515"/>qabaa<Node id="1520"/>,<Node id="1521"/><Node id="1522"/>Akkuma<Node id="1528"/><Node id="1529"/>beekkamu<Node id="1537"/><Node id="1538"/>aaddee<Node id="1544"/><Node id="1545"/>Daraartuu<Node id="1554"/><Node id="1555"/>Tuluu<Node id="1560"/><Node id="1561"/>gara<Node id="1565"/><Node id="1566"/>magaalaa<Node id="1574"/><Node id="1575"/>Amboo<Node id="1580"/><Node id="1581"/>deemte<Node id="1587"/>.<Node id="1588"/><Node id="1589"/>Abbaa<Node id="1594"/><Node id="1595"/>Caalaaf<Node id="1602"/><Node id="1603"/>Abdii<Node id="1608"/><Node id="1609"/>gara<Node id="1613"/><Node id="1614"/>mana<Node id="1618"/><Node id="1619"/>baruumsa<Node id="1627"/><Node id="1628"/>deeman<Node id="1634"/>.<Node id="1635"/><Node id="1636"/>Alamganaa<Node id="1645"/><Node id="1646"/>Boonsaa<Node id="1653"/><Node id="1654"/>bara<Node id="1658"/><Node id="1659"/>kudhasagal<Node id="1669"/><Node id="1670"/>87<Node id="1672"/><Node id="1673"/>dhalate<Node id="1680"/>.<Node id="1681"/><Node

id="1682"/>Inni<Node id="1686"/><Node id="1687"/>kan<Node id="1690"/><Node id="1691"/>jiraate<Node id="1698"/><Node id="1699"/>bara<Node id="1703"/><Node id="1704"/>1948<Node id="1708"/><Node id="1709"/>haga<Node id="1713"/><Node id="1714"/>1967<Node id="1718"/><Node id="1719"/>ti<Node id="1721"/>.<Node id="1722"/><Node id="1723"/>Barsiisaa<Node id="1732"/><Node id="1733"/>Gammachiis<Node id="1743"/><Node id="1744"/>Asfawu<Node id="1750"/><Node id="1751"/>Yunivarsiitii<Node id="1764"/><Node id="1765"/>Adaamaati<Node id="1774"/><Node id="1775"/>qacaramee<Node id="1784"/><Node id="1785"/><Node id="1786"/>hojata<Node id="1792"/>.<Node id="1793"/><Node id="1794"/><Node id="1795"/>Qarshii<Node id="1802"/><Node id="1803"/>1<Node id="1804"/>,<Node id="1805"/>235<Node id="1808"/><Node id="1809"/><Node id="1810"/><Node id="1811"/><Node id="1812"/>qarshii

Appendix C: Sample Afaan Oromo NE Corpus

```
Ooggantuun O
biirichaa O
Oromiyaa B-LOC
Aadde O
Sa'iidaa B-PER
Kadiir I-PER
ibsa O
kennaniiru O
. O

Itti O
dabaluunis O
Aadde O
Sa'iidaan B-PER
hubachiisaniiru O
. O

Balaachoo B-PER
Jabeessaatiin I-PER
. O

Biiroon O
Daldalaa B-ORG
Industirii I-ORG
fi I-ORG
Geejjibaa I-ORG
Oromiyaa I-ORG
beeksise O
. O

Magaalota O
Asallaa B-LOC
fi O
Amboottii B-LOC
leenjiin O
kennamaa O
jira O
. O

Oogganaa O
itti O
aanaan O
Biiroo B-ORG
Daldalaa, I-ORG
Induustirii I-ORG
fi I-ORG
Geejjibaa I-ORG
Oromiyaa I-ORG
Obbo O
Mabraatee B-PER
Gabrayyas I-PER
dhaamsa O
dabarsaniiru O
. O

Balaa O
tiraafikaatiin O
naannoon O
Oromiyaa B-LOC
Itoophiyaa I-LOC
keessaa O
Finfinneetti B-LOC
aanee O
2ffaa O
yoo O
ta u O
itoophiyaan B-LOC
immoo O
biyyoota O
addunyaa O
170 O
keessaa O
sadarkaa O
1ffaarra O
akka O
jirut O
Obbo O
Mabraateen B-PER
himaniiru O
. O

Obbo O
Naasir B-PER
Huseen I-PER
himaniiru O
. O

Godina O
Iluu B-LOC
Abbaa I-LOC
Booraatti I-LOC
magaalaa O
Beddellee B-LOC
bu uuraaleen O
misoomaa O
adda O
addaa O
qarshii O
miliyona B-MISC
17n I-MISC
hojjatamaa O
jiru O
```

. O

Magaalaan O
Beddelee B-LOC
Walgeettii O
daandii O
Jimmaa B-LOC
, O
Mattuufi B-LOC
Naqamtee I-LOC
ta uudhaan O
godinoota O
sadii O
kan O
walitti O
fiddi O
. O

Kantiibaa O
itti'aanaan O
Magaalaa O
Bedellee B-LOC
Obbo O
Faxxanaa B-PER
Maammoo I-PER
akka O
jedhanitti O
, O
mootummaan O
bu uuraalee O
misoomaa O
bajata O
mootummaan O
naannoo O
Oromiyaa B-ORG
ramade O
qarshii O
miiliyoona B-MISC
10fi I-MISC
kuma I-MISC
300n O
hojjetamaa O
jira O
. O

Manni O
barumsaa O
sadarkaa O
2ffaa O
olaanaa O
tokko O
qar. O
miliyoona B-MISC
6 I-MISC
oliin O
ijaaramaa O
jira O
jechuun O
Obbo O
Faxxanaan B-PER
dubbataniiru O
. O

Muudde B-MISC
24 I-MISC
bara O
2002 B-MISC
magaalaa O
Finfinnee B-LOC
galma O
walgahii O
Bulchiinsa O
mootummaa O
naannoo O
Oromiyaatti B-ORG
waan O
adeemsifamuuf O
jira O
. O

Godina O
adaa B-LOC
Oromiyaa I-LOC
naannawa I-LOC
Finfinnee I-LOC
bulchiinsa O
magaalaa O
Sabbataatti B-LOC
qabsooftuu O
Dh.D.U.O B-ORG
duraanii O
kan O
turan O
Aadde O
Tigist B-PER
Kaasaayee I-PER
bulchiinsa O
magaalaa O
Sabbataa B-ORG
wajjiin O
walta uun O
manneen O
jireenyaa O
ijaarsisan O
tibbana O
maatii O
qabsaa otaaf O
kenname O
. O

Bulchiinsi O
magaalaa O
Sabbataas B-ORG
hordoffiifi O
deggersa O
guddaa O
gochuusaa O
kan O
dubbatan O
Obbo O
Abbaaduulaan B-PER
, O
maatiin O
qabsaa otaas O
qabeenyaa O
argatan O
kanatti O
fayadamanii O
ijoolleesaanii O
sirriitti O
kunuunsuu O
, O
guddisuufi O
barsiisuu O
akka O
qaban O
dhaamaniiru O
. O

Aadde O
Tigisti B-PER
Kaasaaye I-PER
bara O
1982 B-MISC
- I-MISC
1987tti I-MISC
qabsooftuu O
Dh.D.U.O B-ORG
turan O
. O

Bara O
1987 B-MISC
booda O
Aadde O
Tigist B-PER

gara O
biyya O
Ameerikaa B-LOC
. O

Maallaqa O
ijaarsa O
manneen O
jireenyaa O
kanaaf O
bahe O
keessaa O
qarshii O
miliyoona B-MISC
1fi I-MISC
kuma I-MISC
400 I-MISC
Aadde O
Tiigist B-PER
kan O
baasan O
yoo O
ta u O
, O
miliiyoona B-MISC
1fi I-MISC
kuuma I-MISC
200 I-MISC
ammoo O
bulchiinsi O
magaalaa O
Sabbataa B-ORG
akka O
ramades O
Aadde O
Tigiist B-PER
dubbataniru O
. O

Itti O
gafatamaan O
waajjira O
DH.D.U.O B-ORG
Obbo O
Zalaalam B-PER
Jamaanee I-PER
. O

Pirezidaantiin O
bulchiinsa O
mootummaa O
naannoo O
Oromiyaa B-ORG
Obbo O
Abbaaduulaa B-PER
Gammadaa I-PER
hubachiisan O
. O

Pirezidaanti O
Abbaa B-PER
Duulaan I-PER
kana O
kan O
hubachiisan O
godinoota O
Harargeetti B-LOC
argaman O
, O
gara O
godina O
addaa B-LOC
Oromiyaa I-LOC
nannawa I-LOC
Finfinneetti I-LOC
baballisuuf O
konfiransii O
ta amerratti O
. O

Qarshii O
kuma B-MISC
17tti I-MISC
gurgurachuunsaaniis
O
dubbatameera O
. O

BBQUO B-ORG
. O

Biiroo B-ORG
Beeksisaafi I-ORG
Quunnamtii I-ORG
Uummataa I-ORG
Oromiyaa I-ORG
. O

Aadde O
Itaafarraawu B-PER
Gonfaatiif I-PER
: O
Himataan O
Bihoon B-PER
Yaalewu I-PER
fi O
himatamtuu O
isin O
jiddu O
kan O
jiru O
gaafa O
18/05/2002 B-MISC
sa'a O
2:30 B-MISC
irratti O
akka O
dhihaattan O
haa O
ta u O
. O

M/M/A/Xiyyoo B-ORG
. O

mana O
murtii O
aanaa O
Xiyyoo B-ORG
. O

Iyyataan O
Obbo O
Haabtaamuu B-PER
Maaruu I-PER
qabeenya O
maqaa O
Obbo O
Araarsoo B-PER
Waaqotiin I-PER
magaalaa O
Roobee B-LOC
ganda O
01 B-LOC
keessatti O
argamu O
gara O
kootti O
haa O
naanna u O
jedheera O
. O

M/M/A/Roobee B-ORG
. O

```
mana O
murtii O
aanaa O
Roobee B-ORG
. O

Obbo O
Alrool B-PER
Eksii I-PER
Seemensiitiif I-PER
: O
Himata O
Obbo O
Sorneessaa B-PER
Ittaanaa I-PER
magaalaa O
Shaashamannee B-LOC
ganda O
```

```
Aleeluu B-LOC
keessatti O
argamu O
irrattti O
waliigalteen O
akka O
digamu O
waan O
gaafataniiru O
. O

M/M/A/Shaashamannee
B-ORG
. O

mana O
murtii O
aanaa O
```

```
Shaashamannee B-ORG
. O

Shaashamannee B-LOC
Aadde O
Sinqinash B-PER
Warqinah I-PER
magaalaa O
Asalla B-LOC
ganda O
03 B-LOC
keessatti O
kaartaa O
manaa O
gateera O
jedhaniiru O
```

Appendix D: Sample Afaan Oromo NE Corpus

Sample Data in Gazetteers for Organization Names Extractor

Complete Organization Names	Dhaabbata Raadiyoof Teleeviziyoona Oromiyaa, Gaazexa Telegiraafi Kooleejii Paaraadayiz Vaalii
Business Types	Warshaa Sukaara Wonji, Giddu-gala Gaba Eleemooqilxu Gorsa Aksiyoona Moordan
Company Preceding Indicator	Kaampaanii

	Warshaa Woldayoota Korp.
Company Preceding Known Part	Gaazexa Muummee Raadiyo
Company following Indicator	Oomisha Damee Daldala gaba
Company following Known Part	Warshaalee PLC Biyoolessaa
Locations	Itiyoophiyaa Adamaa Ameerika Dirree Daawa
Prefix Business	Kaaffee Daldaala Hoteela Maddabira
Postfix Business	Xaa‟o Uffata Qonnaa Teeknooloogii

Sample Data in Gazetteers for Location Names Extractor

Direction	**Baha/Baya** (East) **Lixa** (West) **Kibba** (South) **Kaaba** (North)

City Names	**Adaamaa** (Adama) **Awaasaa** (Awasa) **Aksum** (Axum) **Bahir Daar** (Bahir Dar) **Goondar** (Gonder) **Landan** (London) **Maqalee** (Mekele) **Roobe** (Robe) **Shaashamannee** (Shashemene)
Country Names	**Itoophiyaa** (Ethiopia) **Jarmanii** (Germany) **Inglizii** (England) **Jaappan** (Japan) **Jamaayikaa** (Jamaica) **Biraazil** (Brazil)
State Names	**Affaar** (Afar) **Amaara** (Amhara) **Gaambellaa** (Gambela) **Oromiyaa** (Oromia) **Tigraay** (Tigray) **Finfinnee (Addis Ababa)**
Capital Names	**Finfinnee** (Addis Ababa) **Landan** (London) **Waashingitan** (Washington) **Ankaaraa** (Ankara) **Maadriid** (Madrid) **Paaris** (Paris)
Administrative Divisions	**Rippaaplikii** (Republic) **Bulchiinsa naannoo** (State) **Godina** (Zoni) **Aanaa** (Worade)

	Ganda (Kebele)
Country Pre Indicators	**Rippaaplikii** (The Republic) **Biyyii/Biyyiiti (**TheCountry) **Moottummaa Federaalaa (**The Federal Government) **Dimokraata** (The Democratic) **Moottumma** (The government)
City Preceding Indicators	**Magaalaa** (The City) **Bulchiinsa Magaala**(The City governor) **Gara magaala** (Heading Towards)
City following Indicators	Magaala, naanawa, garaa
Continents	**Afrikaa** (Africa) **Eeshiyaa** (Asia) **Awustiraliyaa** (Australia) **Awurooppaa** (Europe)
Monuments	**Laalibalaa** (Lalibela) **JogolaHarar** (Harar Jugol) **Holqa Soofumer** (Sof Omar Cave)
Mountains	**Ras Dajen** (Ras Dejen) **Tulluu Diimtuu** (Tullu Dimtu) **Gaara Gunaa** (Guna Terara) **Cilaaloo** (Chilalo) **Ammaarro** (Amaro) **Baatuu** (Batu) **Inxooxxoo** (Entoto) **Gaara Mul'ataa** (Gara Muleta)
Rivers	**Abbaay** (Abay) **Gibee** (Gibe) **Takazee** (Tekeze **Waabee** (Wabe) **Gannaale** (Ganale Dorya) **Marab** (Mareb)

	Weebi (Weyib) **Shaballee** (Shebelle) **Awaash** (Awash) **Oomoo** (Omo)
Places	**Gaara** (Mountain) **Dirree/Badhee/Raasaa** (Field)
Oceans, Seas and Islands	**Atlaantikii** (Atlantic) **Iindiyaa** (Indian) **Arkitikii** (Arctic)

Declaration

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

Declared by:

Name: **Abdi SaniGenemo**

Signature: _____

Date: _____

Confirmed by advisor:

Name: Dida Midekso(PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, March 2015.