Using Machine Learning to Improve Rule based Arabic Named Entity Recognition

NERA 2.0

Muhammad Shoaib

Institute of Informatics

The British University in Dubai

A thesis submitted for the degree of

Master of Science (Information Technology)

January 2011

Supervisors: Dr. Khaled Shaalan
and Dr. Sherief Abdallah

Abstract

Arabic Language is widely spoken and highly influential Language both politically and geographically. Thus it is crucial to perform Information Extraction on diverse Arabic texts. In past decade many researchers have targeted the Information Extraction in general and Named Entity Recognition in particular for Arabic language. Mostly researchers have applied Machine Learning for Arabic Named Entity Recognition while few researchers have used hand crafted rules for Named Entity Recognition task.

The Machine Learning techniques and rule based techniques for named entity recognition are mostly viewed as rival approaches. The work presented in this thesis is an effort to combine rule based and Machine Learning approaches into a Hybrid System for Named Entity Recognition. The Person, Organization and Location entities identified by rule based system are used as features combined with several other features for Machine Learning system. The final outcome provides enhanced Named Entity annotations.

The evaluation of the experiments conducted shows that the Hybrid approach stated in thesis significantly improves the quality of named entity recognition of independent rule based system and independent Machine Learning system. Moreover the statistical significance tests confirms that the results obtained are valid and not occurred by chance.

To ...

Scientific Community, keep up the good work,

and ...

My family

Acknowledgements

Firstly I would like to thank Allah the most merciful and kind. I would like to render my gratitude for my thesis supervisors Dr. Khaled Shaalan and Dr. Sherief Abdallah for their guidance, support and ideas throughout the dissertation. They were always there when I needed the help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this Chapter Overview of Named Entity Recognition is given. The Chapter also highlight the research objectives and Scope of this project.

1.1 Overview

Named Entity Recognition is one of the major tasks in Information Extraction, Information Retrieval and Natural Language Processing domains. Named Entity Recognition System is defined as:

"A Named Entity Recognition (NER) system is a significant tool in NLP research since it allows identification of proper nouns in open-domain (i.e., unstructured) text. For the most part, such a system is simply recognizing instances of linguistic patterns and collating them." (Shaalan and Raza, 2008)

As evident from the above definition Name Entity Recognition System identifies the Named Entities such as Person, Location and Organization Names etc. from the text and is usually a preprocessing step for various Natural Language Processing Applications.

1.2 Motivations

Arabic Language is widely spoken and highly influential language. Not only is Arabic an official Language of most of the Gulf countries, Arabic is also the

Language of Holy Quran and thus it has religious affection for Muslims population which is around 1/6th population of the whole world. Since Gulf is rich region in terms of natural resources such as Oil, Natural Gas etc., the Arabic language is also important both geographically and politically. Thus information extraction is crucial when it comes to Arabic language.

Named Entity Recognition is also useful for many tasks other then Information Extraction. Some of them are Machine Translation, Search Results clustering, and Question/Answering etc. (Benajiba et al., 2008*b*). Therefore it is desirable to construct System that can identify Named Entities. The identification of Named Entities can also be used as a preprocessing step for several Natural Language Processing Systems.

The research in Named Entity Recognition for Arabic Language is still in its early phases. Not much has been done for Arabic Natural Language processing in general and Named Entity Recognition in particular. The major reason for this lagging as compared to work in other languages is the lack of available tools and Language Resources for Arabic Language related to Named Entity Recognition task. Moreover Arabic language is highly challenging to deal with when it comes to perform linguistic grammar based processing. Some of these challenges are described and tackled by Shaalan and Raza (2008).

1.3 Goals and Objectives

The major two approaches to perform the Named Entity Recognition includes rule-based techniques and statistical Machine Learning based techniques. Rule based techniques exploits the hand crafted rules that usually maps Linguistic Grammar to regular expression. For example Shaalan and Raza (2008) has worked extensively on rules for Arabic Named Entity Recognition. In statistical techniques the Named entity recognition is treated as classification problem and various features are learned to classify the entities. An example of such system is worked out by (Benajiba et al., 2008*b*). Rule based system has its limitation since it requires an expert linguist to develop rules which can take months. The machine learning systems mostly use language independent features of text. The

main limitation of statistical machine learning based system is that it requires large annotated training data.

The statistical approach and rule based approach for Named Entity Recognition are usually considered as rival approaches. The idea of this research project is to combine both the rule based and Machine Learning approaches into a Hybrid system and study the effects on resultant Named Entity Recognition System. The experiments conducted shows that Hybrid approach can result in higher performance compared with the performance of independent rule based system or independent Machine Learning system for NER task.

The project is developed in pursuit of answers to the question presented in Section 1.4. The project was aimed at developing Named Entity recognition system for the Arabic texts based on Machine Learning techniques that utilizes annotations produced by rule based system. The Machine Learning system runs on the top of Rule based system to enhance the performance of rule based System. The input for the system is Arabic text for which the system will generate Named Entities. Three Named Entity types including "Person", "Organization" and "Location" are implemented and investigated in this research project. These are the most common Named Entity types and are available in most of the annotated corpora.

In order to conduct experiments for evaluating the goals of the project, following components are built. First component of the system is Rule based Named Entity Recognition System built using Gate Developer API. Second component is Machine Learning based system built using Java along with XML parser which server as tokenizer. Front end of the system is developed as Web Interface using Java Server Pages. Various Gazetteers (dictionaries) are collected and enriched semi-automatically to facilitate Rule based and Machine Learning components. For these enrichments various utilities are developed. Finally some utilities are built to transform corpora among different formats.

1.4 Research Questions

The thesis aimed at answering the following research questions:

- Is it possible to improve the performance of rule based system for Named Entity Recognition by applying Machine Learning on the top of rule based system?
- What is the appropriate feature set for Machine Learning to achieve this goal?

Thus the hypothesis of the research can be stated in the form of claim as:

"The Machine Learning applied on the top of rule based system for Named Entity recognition (i.e. Hybrid system) will improve the performance of independent rule based system as well as independent Machine Learning system."

1.5 Organisation of the Thesis

The rest of Thesis is organised as follows. The detailed literature review along with overview of Named Entity Recognition Systems is given in Chapter 2. Chapter 2 also discusses the properties of Arabic language along with challenges faced by researchers in Arabic Natural Language Processing. Chapter 3 describes the Data collection mechanism and resources built for the project. Chapter 4 is devoted to description of rule based system architecture and implementation using Gate Developer IDE. The chapter also discusses the integration of Gate Developer Application with Java Server Pages based web application. Chapter 5 describes the Machine Learning system architecture and implementation along with the overview of software tool WEKA that support Data mining and Machine Learning applications. The details of experiments conducted and results obtained are listed along with the discussion in Chapter 6. Chapter 6 also covers the statistical significance of results obtained. Finally Chapter 7 draws the conclusion along with the direction for future research.

Chapter 2

Named Entity Recognition and Arabic Language - Literature Review

This chapter describes the origin of Named Entity Recognition, applications of Named Entity Recognition, Characteristics of Arabic Language along with the inherent challenges in Natural Language Processing for Arabic Language, and related work done in the filed of Named Entity Recognition for Arabic and other languages.

2.1 Origin of Named Entity Recognition

The concept of Name Entity Recognition was born in Message Understanding Conferences in 1990s. In Sixth Message Understanding Conference (website `http://cs.nyu.edu/cs/faculty/grishman/muc6.html`) held in November 1995, the NER task was formally broken down into three subtasks. These subtasks as described on `http://cs.nyu.edu/cs/faculty/grishman/NEtask20.book_3.html#HEADING4` included

Named Entities - ENAMEX tag To identify proper names including Person, Organization and Location Names.

e.g. <ENAMEX TYPE="LOCATION">North< /ENAMEX>

Temporal Expression - TIMEX tag To identify absolute temporal expressions including Date and Time.

e.g. <TIMEX TYPE="DATE">fiscal 1989< /TIMEX>

Number Expression - NUMEX tag To identify two type of numeric expressions including Money and Percentage.

e.g. <NUMEX TYPE="MONEY">$42.1 million< /NUMEX>

In this research project we consider three Named Entity types including "Person", "Organization" and "Location" which correspond to ENAMEX tag described above.

2.2 Applications of Named Entity Recognition

As described earlier in Chapter 1, the Named Entity Recognition though is an independent task, it can also acts as preprocessing step for various different Natural Language Processing Systems. Some of the application areas that utilizes NER task for preprocessing are discussed below as described by Benajiba et al. (2008*b*); Benajiba and Rosso (2008).

Machine Translation The machine Translation approach for Named Entities is different from approach for normal text. Thus the accuracy of Machine Translation can be increased by using Named Entity Recognition as preprocessing step.

Information Retrieval Given a query from user in natural language, the information Retrieval system retrieves the relevant documents. Since most of the questions are about Named Entities, recognizing them before hand will increase the retrieval of related documents.

Search result clustering is a subtask of text clustering with in Information Retrieval System. The search results of Information Retrieval are clustered to make them easier to read by user.

Question Answering is a system that finds the answer to specific questions of user. Its different from Information Retrieval as it not only has to retrieve the relevant documents but also has to locate where exactly the answers lies with in the documents. Thus Question Answering system utilizes several Natural Language Processing Systems including Named Entity Recognition System.

Apart from abovementioned areas, Named Entity Recognition is also useful for Text Mining, Text to Speech (TTS) Systems, Automatic Summarization, and Speech Recognition Systems etc.

2.3 Arabic Language Characteristics and Challenges

Arabic language is highly challenging to deal with when it comes to perform linguistic grammar based processing. The difference between Arabic script and Roman Script are described by Habash (2010, pg. 15) as:

"Some of the differences, such as script direction, letter-shaping and obligatory ligatures, are effectively abstracted away in computational applications ... The two most prominent differences are perhaps optionality of diacritics and lack of capitalization ... The lack of capital/small letter distinction, which is used in specific ways in different Roman script languages, makes some applications, such named[sic] entity recognition and part-of-speech tagging, more challenging in Arabic."

Some of the inherent challenges in Arabic Natural Language Processing as described by Shaalan and Raza (2008, 2009) are discussed briefly in following sections:

2.3.1 Complex Morphology

Arabic is highly inflected language. Word are formed using stem or root, with prefixes and suffixes characters. As shown in Figure 2.1 the Arabic word "وكتبهم" (transilterated as wa-kutub-hum) is formed using prefix "و" (wa), stem word

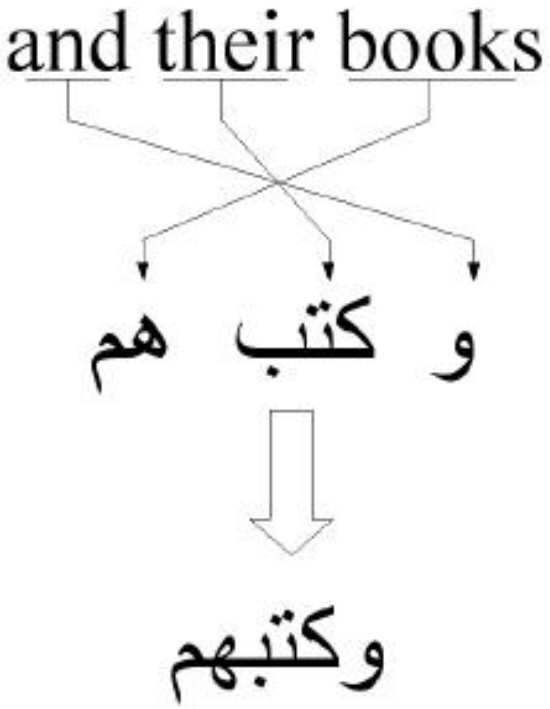

Figure 2.1: Example Arabic Word Formation - (Benajiba and Rosso, 2007)

"كتب" (kutub) and suffix "هم" (hum). "This concatenative strategy to form words in Arabic causes data sparseness; hence this peculiarity of the Arabic language poses a great challenge to NER systems." (Shaalan and Raza, 2008).

2.3.2 Lack of Capital Letters

Arabic language lacks the capital letters and thus other heuristics have to be applied for detecting Named Entity boundaries such as preceding or succeeding indicator words as applied by Shaalan and Raza (2008, 2009).

2.3.3 Non Standard Written Text

The translated and transliterated words to Arabic are not standardized. This is problematic as most of the time all possible spelling variants are not possible to take into consideration.

2.3.4 Ambiguity and lack of Diacritization

The written Arabic lacks the Diacritics (short vowels). Attia et al. (2010) outlines the issue of diacritization as:

"As most Arabic texts that appear in the media (whether in printed documents or digitalized format) are undiacritized, restoring diacritics is a necessary step for various NLP tasks that require disambiguation or involve speech processing."

Missing diacritics are not the only problem. The Arabic words can have different meanings in different contexts which increases the complexity of Named Entity Recognition Systems.

2.3.5 Lack of Resources

The lack of resources for Arabic Named Entity Recognition task is the major reason of research in this domain being still in its infancy. Most of the available resources are either very costly or are of low quality. Thus researchers have to build up their own resources for training and evaluation of Arabic Named Entity Recognition systems. The lack of using standardized resources creates problem of comparing performance among different systems.

2.4 Rule Based Systems

In this section literature review of rule based systems is presented. Rule based systems exploits the hand crafter rules for Named Entity Recognition task. The rule based systems require extensive work from expert linguists and thus can result in near human accuracy. Rule based systems usually target single language only because of huge difference among grammars of different language. Only few researchers have used hand crafted rules to tackle Named Entity Recognition task for Arabic. The rules within rule based systems are implemented as regular expressions or finite state transduction based grammar for pattern matching. These rules mostly rely on large lists of lookup gazetteers which is major shortcoming of rule based systems.

2.4.1 Related Work

TAGARAB (Maloney and Niv, 1998) is one of the early systems that uses rule based pattern matching for Named Entity Extraction in Arabic. TAGARAB uses morphological analysis of text in conjunction with pattern matching to achieve higher accuracy as compared to simple pattern matching.

Traboulsi (2009) has discussed the application of local grammar based approach in domain of Arabic Language. The grammar was extracted by applying corpus analysis over range of untagged Arabic corpora. The result of the research by Traboulsi (2009) is a finite state automata to extract named entities from Arabic text.

Person Named Entity Recognition for Arabic (PERA) developed by Shaalan and Raza (2007) utilizes hand crafted grammar rules in conjunction with Whitelist dictionaries to extract person names from Arabic text. The filter mechanism is applied as last stage of PERA to omit the incorrect entities extracted by Whitelist or grammar.

Named Entity recognition for Arabic (NERA) (Shaalan and Raza, 2008, 2009) is an extension of PERA (Shaalan and Raza, 2007). NERA is a hand crafted rule based system that utilizes Whitelist and exploits finite state transduction based grammar to identify ten types of Named Entities. Filter mechanism is applied using blacklist dictionaries to omit the incorrect entities identified. The supported entities by NERA, includes Person, Location, Organization, Date, Time, Price, Measurements, Phone Number, ISBN, and File Names. NERA is the only system of its kind so far that can extract ten types of Named Entity for Arabic Language with high precision and recall.

Elsebai et al. (2009) also utilizes hand crafter rules to identify Person names in Arabic text. Elsebai et al. (2009) claim better performance in term of F-Measure over PERA system developed by Shaalan and Raza (2007) despite the fact that they used different corpora for evaluations.

Riaz (2010) has worked out rule based system for Urdu language. Although Urdu uses Arabic script and some vocabulary from arabic as well, yet the systems developed for Arabic Language are not useful for Urdu natural language processing because of different grammars as discussed by Riaz (2010).

2.5 Machine Learning Based Systems

Machine Learning is the mostly applied method for Named Entity Recognition for all major languages including Arabic. Named Entity Recognition is viewed as classification problem for applying Machine Learning. The Machine Learning technique exploits the features of text to classify them as particular Named Entity or as normal text. The features can include both language specific features (e.g. Part of Speech information, Morphological features etc.) and language independent features (e.g. length of the word etc.). A better performance is achieved by using mix of both language dependent and language independent features. The advantage of using Machine Learning is that extensive knowledge of target language is not required thus omitting the need of expert linguists. Moreover Machine Learning system built for one domain or language can easily be modified to fit other languages or domain unlike rule based systems. Major shortcoming of Machine Learning approach is requirement of large corpora with annotated text for Named Entities.

The most commonly published Machine Learning approaches for Named Entity Recognition includes Conditional Random Field (CRF), Hidden Markov Model (HMM), Support Vector Machine (SVM), Maximum Entropy (ME), Decision Trees, etc. The book "Introduction to Machine Learning"(Alpaydin, 2004) is good resource covering major Machine Learning approaches along with the underlying theory.

2.5.1 Related Work

One of the early attempts to utilize Machine Learning for Named Entity Recognition is published by Baluja et al. (2000). English was the target Language for Named Entities identification. The features used for Machine Learning belonged to Word Level Features, Dictionary Look-Up, Part of Speech Tags and Punctuation. Baluja et al. (2000) reported accuracy of system comparable to state-of-the-art rule based systems of that time.

ANERSys(Benajiba et al., 2007) is Named Entity Recognition System based on Maximum Entroy. The authors Benajiba et al. (2007) also developed resources as part of the research. These resources include Annotated Corpus for Named

Entity Recognition called ANERcorp and set of gazetteers called ANERgazet. The baseline results was acquired by assigning each word in Test set, a class that was most frequently assign to it in Training set. Later the training and testing was done using Maximum Entropy approach. The authors reported significant improvement over baseline results.

The work of ANERSys was extended to ANERSys 2.0 by Benajiba and Rosso (2007). The approach was to use Maximum Entroy along with Part of Speech information. The same baseline was used as in ANERSys. The authors (Benajiba and Rosso, 2007) reported significant improvement over baseline results, results from ANERSys and results from demo version of Siraj (Sakhr) which is a commercial system for Named Entity Recognition.

The Benajiba and Rosso (2008) employed Conditional Random Fields instead of Maximum Entropy for ANERSys system. The results were improved further from the results stated by Benajiba and Rosso (2007) for ANERSys 2.0.

Abdul Hamid and Darwish (2010) also used Conditional Random Fields on simplified feature set for Arabic Language. As per the Abdul Hamid and Darwish (2010), the most important features were leading and trailing character n-grams in words. By virtue of these simplified features Abdul Hamid and Darwish (2010) claimed that morphological or syntactic analysis and gazetteers are not needed. The Abdul Hamid and Darwish (2010) reported the improvement over others related work.

Mayfield et al. (2003) have used Support Vector machine for English and German Language. The interesting fact is that Mayfield et al. (2003) used hundreds of thousands of language independent features for training and testing. The Mayfield et al. (2003) suspects that there is some degree of inherent over fitting but the effect is not large and the approach can be easily utilized on different languages. The major shortcoming of this approach is extremely slow performance because of large number of feature for training and testing.

Ekbal and Bandyopadhyay (2009) used Voted Named Entity Recognition System which is combination of different Machine Learning Classifier including Conditional Random Fields (CRF), Maximum Entroy (ME) and Support Vector Machine (SVM). The features used were consisting of both language dependent and language independent features. The system was tested on Bengali language and

Ekbal and Bandyopadhyay (2009) showed that language dependent features can improve the accuracy with great margins. Ekbal and Bandyopadhyay (2010*a*) also describes the similar approach with the difference that unlabelled data is used which reduces the need of annotated corpus for training.

Ekbal and Bandyopadhyay (2010*b*) has applied SVM on Bengali and Hindi Languages. They used only language independent features for training and testing. Very High Recall and Precision is reported by Ekbal and Bandyopadhyay (2010*b*).

Support Vector Machine is also employed by Benajiba et al. (2008*a*) for Machine Learning based system. Benajiba et al. (2008*a*) studied the impact of different features on performance. They further reported that best performance was obtained using all the features. The domain of features includes Contextual, Lexical, Gazetteers, Morphological, Part Of Speech, Nationality, and Corresponding English Capitalization.

Benajiba et al. (2008*b*) used Support Vector Machine (SVM) and Conditional Random Fields (CRF) to find optimized set of features and compare results from two approaches. The features were quite similar to those used in Benajiba et al. (2008*a*). Benajiba et al. (2008*b*) reported that performance of SVM and CRF are quite similar except for the fact that SVM perform more robustly on data with random contexts.

Benajiba et al. (2010) have used parallel corpus for English/Arabic and used bootstrapping to extract noisy features. The noisy features are used in conjunction with gold standard features. Benajiba et al. (2010) reported the improvement on overall performance of Named Entity Recognition.

Mao et al. (2007) reported that Machine Learning Systems for Named Entity recognition faces lower recall. The reason for lower recall is dominance of “None” class i.e. the words that does not belong to any Named Entity class dominates the words that appears as Named Entities. Mao et al. (2007) thus recommends several non-local features for words that effectively improve the overall recall of Machine Learning System.

Zhang et al. (2004) has applied Statistical Machine Learning approach to retrieve what the authors referred to as ”Focused Named Entities”. Zhang et al.

(2004) define the Focused Named Entities as the Entities which are most topical in given document and are useful in areas such as document summarization, search results ranking, entity detection and tracking etc. Zhang et al. (2004) experimented with different features and applied there approach on Chinese language. Zhang et al. (2004) claimed achievement of near human accuracy for Focused Named Entity Detection on Chinese language.

2.6 Hybrid Systems

Hybrid approach is the combination of hand crafted rule based system and Machine Learning system. The method described in this thesis also belongs to Hybrid approach since we use annotations provided by Rule Based Systems as features for Machine Learning system. The Hybrid approach can also be used other way round i.e. by applying Rules on the annotation provided by Machine Learning based System. The second approach is very rare to best of writer's knowledge. Hybrid approach should not be confused with Hybrid Machine Learning approach which utilizes more then one Machine Learning techniques. The related work is presented below.

2.6.1 Related Work

One of the early attempts to utilize Hybrid approach for Named Entity Recognition was done by Srihari et al. (2000). The approach was to combine hand crafted rule based grammar with two Machine Learning approaches namely Hidden Markov Model (HMM) and Maximum Entropy Model (MaxEnt). Very high precision of Named Entity Tagging is reported by Srihari et al. (2000).

Nguyen and Cao (2008) employed Hybrid approach to disambiguate the candidate named entities. Nguyen and Cao (2008) incrementally apply there approach which comprised of two stages. First stage is to identify candidate Named Entities using patterns and heuristics while in second stage vector space model is used to rank the candidates. Nguyen and Cao (2008) reports high accuracy and claimed that the approach can be used to construct robust Named Entity disambiguation system.

Mencius (han Tsai et al., 2003) is Named Entity recognition system for Chinese Language based on Hybrid approach. InfoMap, a template matching tool is incorporated in Maximum Entropy framework. The template representations of Named Entities in InforMap are used as features for Maximum Entropy. Accuracy of the Mencius is reported to be better then rule based system independently and Machine Learning based System alone. han Tsai et al. (2003) used character based tagging to avoid errors caused by word segmentation.

Biswas et al. (2010) used Hybrid approach and applied Maximum Entropy (ME) with Hidden Markov Model (HMM) followed by rules to detect Named Entities. The rules were used to detect entities including numbers, measures and time etc. The approach was applied on Oriya Language and Biswas et al. (2010) reported high accuracy for documents with different domains including Science, Arts, World Affairs and Commerce.

The Hybrid approach is also applied on Portuguese Language by Ferreira et al. (2007). Ferreira et al. (2007) uses rules for numbers, measures, time and addresses as these entities have fixed structures while Hybrid approach is applied for proper names.

The other relevant work to the approach described in this thesis is worked out by Petasis et al. (2001). Petasis et al. (2001) have used Machine Learning to maintain rule based system by using output of rule based system as features for Machine Learning. In the first stage the authors trained the Model for machine learning by using annotations of Rule Based System. In the next stage they applied independently the Rule Based System and the Machine Learning Model on unseen data. The cases of disagreements within rule based system and Machine Learning system are presented to expert Linguist. The Linguist can identify the problems in recognition by considering cases of mismatch and the rules within rule based system can be maintained to cater these recognition problems. The stages are shown in Figure 2.2. The major difference between our method and approach of Petasis et al. (2001) is that Petasis et al. (2001) only finds mismatches between rule based system and Machine Learning based system and cases of mismatch are presented to expert Linguist. There is no tagged corpus utilized in training phase of Machine Learning Model. Moreover results of rules based system are not used as features for Machine Learning which differentiate it with our research.

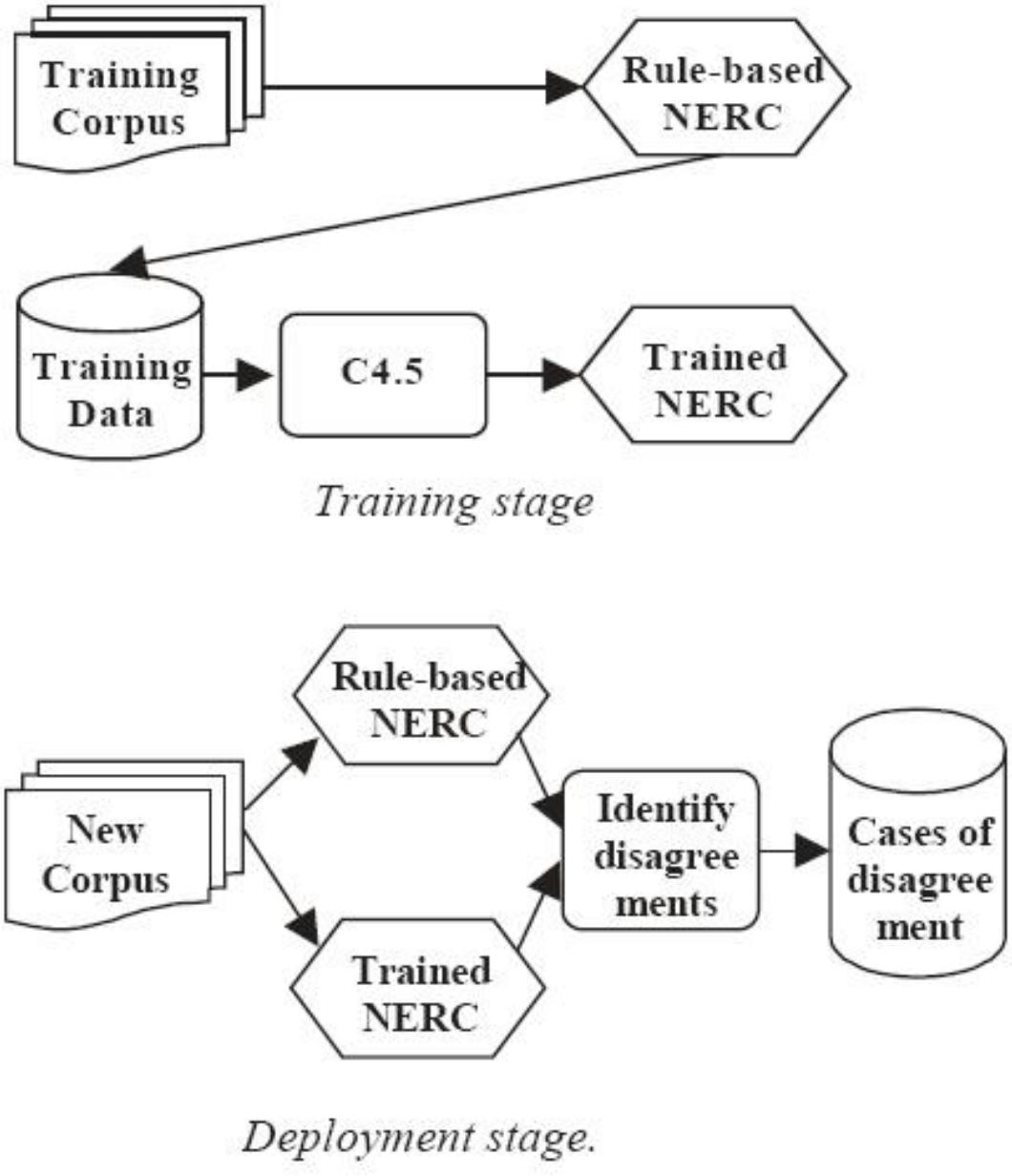

Figure 2.2: Example Hybrid System - (Petasis et al., 2001)

The other issue with the approach described by Petasis et al. (2001) is that Linguist have to manually locate reason with particular mismatch case within the Classification (from Machine Learning System) or Recognition (from Rule Based System).

To best of our knowledge there is no available system for Arabic Language based on Hybrid approach. Thus the approach presented in this thesis is first of its kind for Arabic Language.

2.7 Chapter Summary

The chapter describes the Literature review for Named Entity Recognition and Arabic Language. The Origin and Applications of Named Entity Recognition are given along with the related work done in Named Entity Recogniton for all three major approaches including rule-based, Statistical Machine Learning based and Hybrid approaches. The Arabic Language characteristics and Challenges in Arabic Natural Language Processing are also described in this chapter. The Data collection mechanism and resources build for the research project are described in the next chapter.

Chapter 3

Data Collection

In this Chapter Data collection mechanism is described in detail. The chapter also discusses the resources built for project. The data is required for the implementation of Rule based and Machine Learning based system.

3.1 Data Collection Methodology

The Data required for project includes Various Gazetteers (Dictionaries) and annotated Corpora for Named Entity Recognition Task. The Gazetteers are crucial component of any Named Entity Recognition system especially for Person, Location and Company Name extractors. Annotated Corpora on the other hand are required for evaluating the performance of the system. The annotations in annotated Corpora are also necessary for Machine Learning system for Training and testing of the data. The Data Collected is in accordance with the specification of NERA system and the mechanism of Data collection was similar to the schemes described by Shaalan and Raza (2008) for NERA System.

3.1.1 Resources Used

The three major sources that were used for data collection are described in this section. The technical reports for NERA explained few examples along with Sample data entries which were directly incorporated gazetteers build for NER task. This data although minimal was helpful to acquire correct type of words

for different gazetteers. The Gate Developer IDE developed by Cunningham et al. (2002) provides plug-in for Arabic Named Entity Recognition. This plug-in contains various Gazetteers which are useful for Named Entity Recognition Tasks. These Lists were processed and Data was extracted for our Rule based and Machine Learning system as described in subsequent sections. Furthermore List of Person Names, Locations and Organization Names prepared by Yassine Benajiba were also used. These Lists are available for download from the URL `http://www1.ccls.columbia.edu/~ybenajiba/downloads.html`. Besides these resources various different websites were also used for data collection.

3.2 Training and Reference Corpora

Corpora (singular corpus) 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. Two Corpora were used extensively for project listed as under:

1. The ACE 2003 Multilingual Training Set[1]
2. ANERcorp Corpus Prepared by Yassine Benajiba[2].

These corpora were used for Data Acquisition, evaluation of systems and as a Training/Testing data for Machine Learning System.

3.2.1 Overview of ACE 2003 Multilingual Training Set

ACE stands for Automatic Content Extraction, a technology build to support automatic processing of human language in text form[3]. ACE 2003 Multilingual Training Set corpus is distributed by Linguistic Data Consortium (LDC) under the Catalog number LDC2004T09 and ISBN 1-58563-292-9. ACE provides several different files in Standard Generalized Markup Language (SGML) format. These

[1] Available to BUID under Licence

[2] Available for download from `http://users.dsic.upv.es/~ybenajiba/`

[3] http://www.itl.nist.gov/iad/mig/tests/ace/

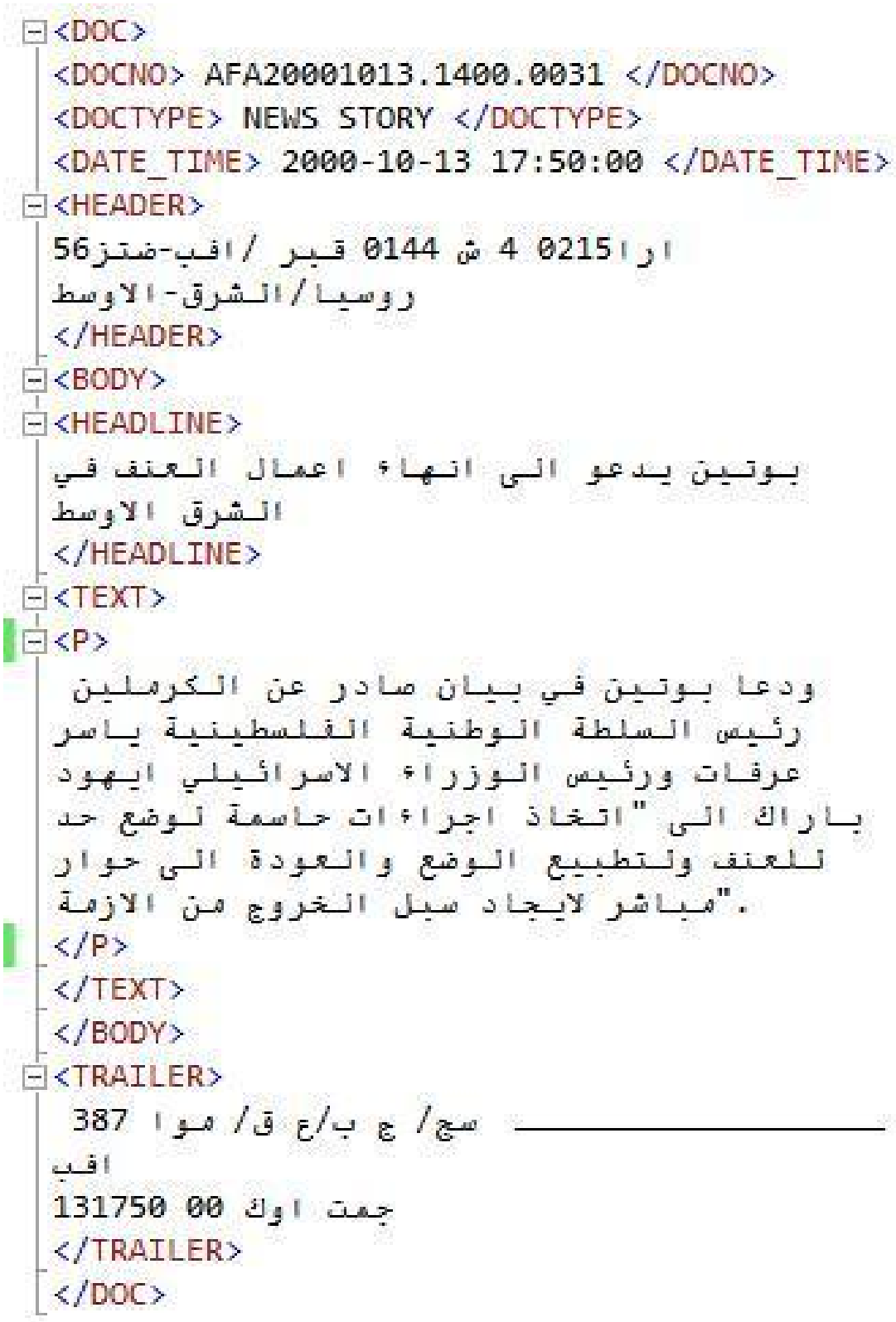

Figure 3.1: ACE Data - Sample

```
<?xml version="1.0"?>
<!DOCTYPE source_file SYSTEM "ace-rdc.v2.0.1.dtd">
<source_file URI="AFA20001013.1400.0031.sgm" SOURCE="newswire"
TYPE="text" VERSION="2.0" AUTHOR="LDC" ENCODING="UTF-8">
  <document DOCID="AFA20001013.1400.0031">
    <entity ID="AFA20001013.1400.0031-E20">
      <entity_type GENERIC="FALSE">ORG</entity_type>
      <entity_mention ID="20-38" TYPE="NAME" REFERENCE="INTENDED">
        <extent>
          <charseq>
            <!-- string = "الكرملين" -->
            <start>356</start>
            <end>363</end>
          </charseq>
        </extent>
        <head>
          <charseq>
            <!-- string = "الكرملين" -->
            <start>356</start>
            <end>363</end>
          </charseq>
        </head>
      </entity_mention>
      <entity_attributes>
        <name>
          <charseq>
            <!-- string = "الكرملين" -->
            <start>356</start>
            <end>363</end>
          </charseq>
        </name>
      </entity_attributes>
    </entity>
  </document>
</source_file>
```

Figure 3.2: ACE Entity Information - Sample

files contain data from Broadcast News and News Wire Articles. Each data file in ACE corpus has corresponding XML file which provides Entity information for words in data file. The Entity types covered by ACE 2003 Data includes Person, Organization, Location, Facility and Geo Political Entity (GPE). A sample data from ACE is listed in Figure 3.1 and Entity information for this data file is listed in Figure 3.2.

3.2.2 Overview of ANERcorp

ANERcorp is a corpus prepared by Yassine Benajiba for Named Entity Recognition Task in Arabic Language. With more then 150,000 words annotated for

```
فرانكفورت B-LOC
د) O
ب O
أ) O
أعلن O
اتحاد B-ORG
صناعة I-ORG
السيارات I-ORG
في O
ألمانيا B-LOC
امس O
الاول O
أن O
شركات O
صناعة O
السيارات O
في O
ألمانيا B-LOC
```

Figure 3.3: ANERcorp Corpus - Sample Data

Named Entity Recognition, ANERcorp is ideal for Machine Learning based system as large annotated text is required for better Machine Learning. The details of ANERcorp corpus along with parsing information is described in (Benajiba et al., 2007) and (Benajiba and Rosso, 2007). The ANERcorp is easy to parse as each line contains single word with its Entity Information. The corpus is tagged in CONLL format as shown in Figure 3.3. The possible entity information attached to each tag as described in is listed 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

3.2.3 Transformation of Corpora

In order to utilize Corpora described in previous sections, we transformed them into XML format using JAVA code. Only Person, Organization and Location entities are taken into consideration from source corpora during transformation, while other entity types are ignored. The XML format is in compliance with NERA system specification and can also be used in GATE Developer. For ACE Training set all the files were parsed and transformed into two XML files, one for Broadcast News data and other for Newswire data. All the data of ANERcorp was transformed into single XML file. A sample transformed XML file is demonstrated in Figure 3.4.

3.3 Gazetteers for Person Names Extractor

The Gazetteers built for rule based Person Names extractor are described below. Three sample entries from each gazetteer are listed in Table 3.1 along with their transliterations in English.

<NE>
<Paragraph>
أعلن (أ ب د) <Location>فرانكفورت</Location>
في <Organization>السيارات صناعة اتحاد</Organization>
في السيارات صناعة شركات أن الاول امس <Location>ألمانيا</Location>
السوق ركود ظل في صعبا عاما تواجه <Location>ألمانيا</Location>
ملايين خمسة حوالي الانتاج يبلغ لان تسعي وهي والصادرات الداخلية
الاتحاد رئيس وقال . 2002 عام في سيارة
للاتحاد سنوي تقرير آخر إعلان عند <Person>جوتشولك برند</Person>
الخطوط إلي يفتقر مازال السوق مستقبل إن
أنه قال أنه من الرغم وعلي . الواضحة
غير من يبدو فإنه العام هذا مرتفع مستوي عند السيارات صادرات تظل أن يتوقع
صادرات زادت عندما 2001 عام نموها سجل مستوي إلي تصل أن المحتمل
المائة في ستة بنسبة الركاب سيارات
<Person>جوتشولك</Person> ورأي . سيارة مليون 6.3 إلي لتصل
العام هذا السيارات لصادرات الاجمالي الحجم يبلغ أن يتعين أنه
. سيارة مليون 4.3 حوالي
هو مثلما بالفعل جار لعام بالنسبة التكهن الصعب من كان ما نادرا قائلا وأضاف
<Location>ألمانيا</Location> في الجديدة السيارات عدد يصل أن يتعين أنه وقال . الحالالان
الماضي العام في سيارة مليون 34.3بـ بالمقارنة سيارة مليون 2.3 حوالي إلي العام هذا
في السيارات صناعة شركات بأن تكهن <Person>جوتشولك</Person> أن من الرغم وعلي .
<Location>ألمانيا</Location>
الماضي العام إنتاج مستوي عن يقل هذا فإن القادمة عشر الاثني الشهور في سيارة ملايين خمس حوالي تنتج سوف
ركاب سيارة مليون 3.5 بلغ الذي
تاريخ في نتيجة أفضل ثالث كان 2001 عام إنتاج مستوي أن وقال .
في سيارات صناعة شركات أهم
الخارجية والسياسة للأمن الأعلى الممثل رجح . <Location>ألمانيا</Location>
<Location>الأوروبي</Location>
الأسبوع <Organization>الأمن مجلس</Organization> يتوصل أن <Person>سولانا خافيير</Person>
المقبل
الاعتداء عن الناشبة للأزمة حل إلي
المتواصل <Location>الإسرائيلي</Location>
في صحفي مؤتمر خلال وقال . <Location>لبنان</Location> على أيام ثمانية منذ
إن <Person>مبارك حسني</Person> الرئيس لقائه بعد <Location>القاهرة</Location>
<Organization>الأمن مجلس</Organization>
</Paragraph>
</NE>

Figure 3.4: Transformed Corpus - ANERcorp Data

Table 3.1: Sample Data in Gazetteers for Person Names Extractor

Complete Names	حسن نصر الله	محمد سعيد	كوفي أنان
	Hassan Nasar Allah	Muhammad Saeed	Kofi Anan
First Names	عبدالله	عمر	إسحق
	Abdullah	Umar	Ishaq
Middle Names	العزيز	سلمان	سعيد
	Al Aziz	Salman	Saeed
Last Names	المجيد	صالح	مبارك
	Al Majeed	Salah	Mubarak
Honorifics	ألسلطان	السيد	الملك
	Al Sultan	Al Syed	Al Mulk
Person Titles	السيدة	الشيخ	فضيلة الشيخ
	Al Syedda	Al Sheikh	Fazila Al Sheikh
Job Titles	الجراح	الدكتورة	الرسام
	Al Jarrah	Al Ductora	Al Rassam
Locations	أثيوبيا	اليابان	أريزونا
	Ethopia	Al Yaban	Arizona
Numbers	الثالث	الرابع	الأول
	Al Salis	Al Raabea	Al Awwal

Continued on next page

Person Indicator	المشرف الرياضي	الأصولي	رئيس الوزراء
	Almusharaf Alriyazi	Al Usoli	Rais Al Wazral
Laqabs	الازهر	الأسد	الأمين
	Al Azhar	Al Asad	Al Amin

Complete Names containing Complete Names of Persons

First Names containing First Names of Persons

Middle Names containing Middle Names of Persons

Last Names containing Last Names of Persons

Honorifics containing Honorifics for Persons

Person Titles containing Persons Titles

Job Titles containing Job Titles such as Doctor in English.

Locations containing Locations for Person Names.

Numbers contains Number usually appearing in the Names of the Kings and Rulers.

Person Indicator containing Person Indicators.

Laqabs containing Laqabs indicating description of the persons.

3.4 Gazetteers for Organization/Company Names Extractor

The Gazetteers built for rule based Organization Name extractor are described below. Three sample entries from each gazetteer are listed in Table 3.2 along with their English translations.

Table 3.2: Sample Data in Gazetteers for Organization Names Extractor

Complete Organization Names	سكاي نيوز	نيوزويك	سوبر ماكس
	Sky News	News Week	Super Max
Business Types	شركة البترول الوطنية	الخدمات الطبية	شركة الاستشارات
	National Oil Company	Medical Services	Consultancy Firm
Company Following Indicator	فرع	وشركاه	ش م م
	Branch	and Companies	LLC
Company Following Known Part	للأخبار	جورنال	انترناشيونال
	News	Journal	International
Company Preceding Indicator	اللوازم	صحيفتي	بورصة
	Supplies	Newspaper	Stock Exchange
Company Preceding Known Part	شركة	راديو	الإدارة
	Companys	Radio	Department
Locations	آسيا الوسطى	اليابان	أثينا
	Central Asia	Japan	Athens
Prefix Business	الزراعية	التجارية	الصناعية
	Agricultural	Commercial	Industrial

Complete Organization Names List containing Complete Names of Organizations

Business Types List containing Business Types

Company Following Indicator List containing Following indicator words for Organizations

Company Following Known Part List containing Following known Parts for Organizations

Company Preceding Indicator List containing preceding indicator words for Organizations

Company Preceding Known Part List containing preceding known Parts for Organizations

Locations List containing Locations

Prefix Business List containing Prefix words for different businesses

3.5 Gazetteers for Location Names Extractor

The Gazetteers built for rule based Location Name extractor are described below. Three sample entries for each gazetteer are listed in Table 3.3 along with their transliteration in English.

Table 3.3: Sample Data in Gazetteers for Location Names Extractor

Direction1	شمال	شرق	غرب
	North	East	West
Direction2	الشمال	الشرق	الغرب
	The North	The South	The West
Direction3	شمالي	شرقي	غربي
	In the North of	In the East of	In the west

Continued on next page

Direction4	الشمالي	الشرقي	الغربي
	The Northern	The Eastern	The Western
Direction5	الشمالية	الشرقية	الغربية
	The Northern	The Eastern	The Western
City Names	مسقط	الطائف	شيكاغو
	Muscat	Taif	Chicago
Country Names	فلسطين	استراليا	باكستان
	Phalestine	Australia	Pakistan
State Names	شارجة	تكساس	نيويورك
	Sharjah	Texas	New York
Capital Names	طوكيو	كراتشي	نيودلهي
	Tokyo	Karachi	New Delhi
Administrative Divisions	جمهورية	ولاية	مملكة
	Republic	State	Kingdom
Country Preceding Indicators	الاراضي المحتله	جمهورية	مملكة
	Occupied Territories	Republic	Kingdom
Country Post Indicators	الفيديرالية	المتحدة	الديمقراطية
	Federal	United	The Democratic

Continued on next page

City Preceding Indicators	متوجهة الى	مطار	مدينة
	Heading Towards	Airport	City
City Post Indicators	عاصمة المالية	العاصمة الشتوية	العاصمة
	Financial Capital	Winter Capital	The Capital
Continents	أسيا	أفريقيا	أوروبا
	Asia	Africa	Europe
Monuments	قوس النصر	برج ايفل	بيكادللي
	Triumphal arch	Eiffel Tower	Piccadilly
Mountains	جبال لبنان الشرقية	جبال أراراط	جبل احد
	The mountains of eastern Lebanon	Mount Ararat	Uhad Mount
Rivers	نهر الفرات	نهر دجلة	نهر اليرموك
	Farat River	Dajla River	Alyarmouk River
Places	سوق	جبل	ميدان
	Market	Mountain	Field
Oceans, Seas and Islands	بحر الصين	خليج المكسيكو	جزر سليمان
	China Sea	Mexican Bay	Solomon Island

Direction1 List of directions in their primitive form e.g. "شمال".

Direction2 List of directions in their definite form e.g. "الشمال الشرقي" or "الشمال".

Direction3 List of directions with suffix "ي" e.g. "شمالي" indicating relative position.

Direction4 List of directions in their definite form with suffix "ي" e.g. "القطاع الشمالي" indicating relative position.

Direction5 List of directions in their definite form with suffix "ية" e.g. "المنطقة الشمالية" indicating relative feminine position.

City Names List of City Names.

Country Names List of Country Names.

State Names List of State Names.

Capital Names List of Capital Names.

Administrative Divisions List of Administrative Divisions.

Country Preceding Indicators List of words preceding Country Names.

Country Post Indicators List of words that usually follows Country Names.

City Preceding Indicators List of words preceding City Names.

City Post Indicators List of words that usually follows City Names.

Continents List of Continents.

Monuments List of Monuments.

Mountains List of Mountains.

Rivers List of Rivers.

Places List of Places.

Oceans, Seas and Islands List of Oceans, Seas and Islands.

Miscellaneous Miscellaneous Lists.

3.6 Chapter Summary

In this chapter details of Data Collection mechanism and resources built for the project are described. The overview of two reference Corpora used for project along with Sample Data are described. Transformation of Corpora into XML format required by project is also described briefly. The gazetteers built for NER task along with sample data are also discussed in detail. The implementation of rule based system in detail is described in next chapter.

Chapter 4

Implementation of Rule Based Named Entity Recognition System

In this Chapter architecture and implementation of NERA rule based system is described along with the overview of development platform "Gate Developer IDE". The chapter also explains the integration of NERA rule based system with Java server pages based web application.

4.1 Overview of Gate Developer IDE

The Gate Developer (Cunningham et al., 2002) is an IDE that facilitates the development of Natural Language processing systems. The documentation for Gate Developer along with samples and tutorials is available at `http://gate.ac.uk/gate/doc/`. The Gate Developer IDE supports components based development. The components are referred to as CREOLE which is acronym for "Collection of REusable Objects for Language Engineering". The most important resources in Gate Developer are Language Resources and Processing Resources. The Language Resources are Corpora or documents that include text with optional annotations. Processing Resource are the units of application such as Tokenizer, Java Annotation Patterns Engine (JAPE), Gazetteers etc. Gate Applications are

pipeline of Processing resources. The processing resources are run over Language resources to provided annotations.

The Gate Developer IDE also provides plug-in for Arabic Named Entity Recognition which we will refer to as "Gate Entity Recognition for Arabic" (GERA). The GERA system rely on lookup gazetteers and provides Named Entity Annotations for Arabic text including Location, Person, Organization, Money, Percentage etc.

4.2 Overview of NERA System

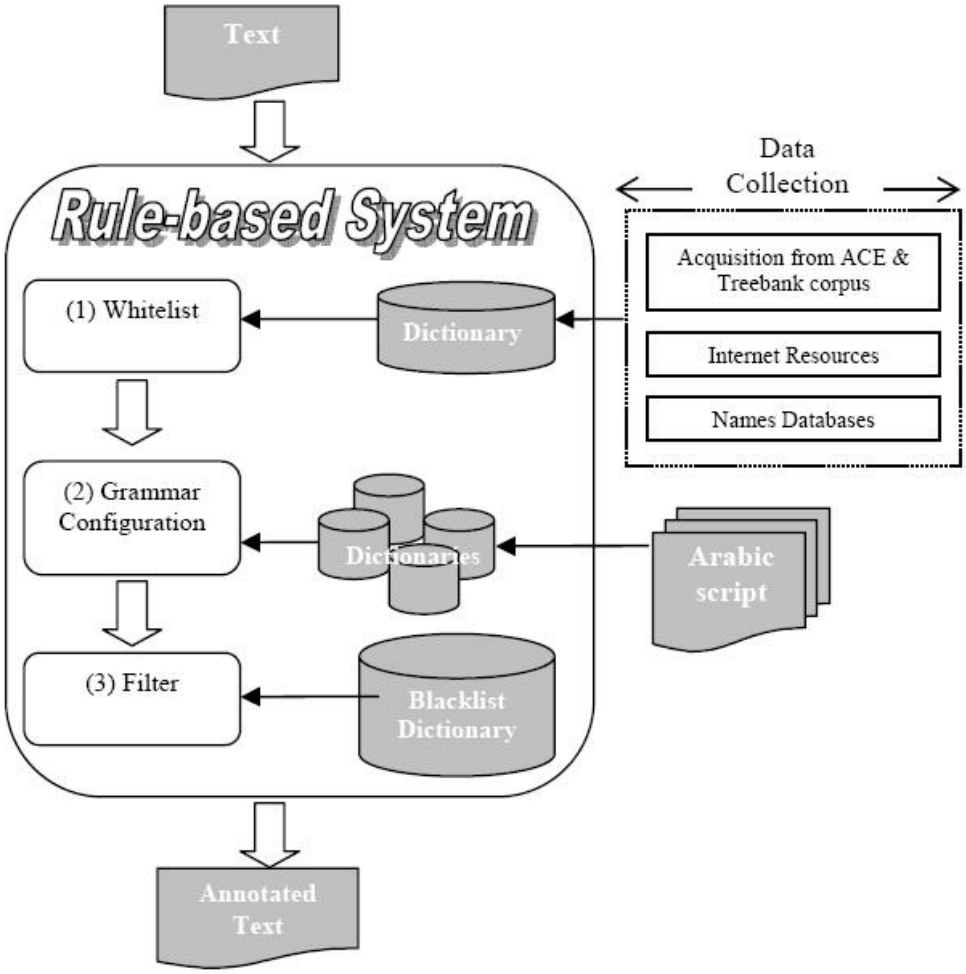

Figure 4.1: NERA System Architecture - (Shaalan and Raza, 2008)

The NERA system is rule based system for Arabic Named Entity Recognition developed by Shaalan and Raza (2008). The NERA system was developed using FAST ESP platform and was incorporated into Fast Search Engine. Fast Search Engine is a commercial tool that is acquired by Microsoft in 2008[1]. The architecture of NERA system is given in Figure 4.1. The three main components of NERA are described in following sections.

4.2.1 Whitelist

The Whitelists are dictionaries of Named Entities that are matched with target text irrespective of the rules. These Whitelists are referred to as Automatons with in FAST ESP platform. The exact matches of target text with Whitelist dictionary entries are reported as Named Entities. The component for matching text with Whitelist dictionaries is called Verbatim matcher in FAST ESP platform.

4.2.2 Grammar Configuration

The Grammar Configuration consists of Pattern matching rules. The rules are based on regular expressions and utilizes several different dictionaries within the rules as shown in Figure 4.1.

4.2.3 Filter

Filtration is performed in the last stage of Named Entity Recognition in NERA system. The filtration is used to omit the incorrect words being recognised in earlier phases as Named Entities. The filtration is performed using Blacklist dictionaries containing entries which should be rejected as Named Entities.

4.3 Implementation of NERA

The reproduced NERA System is used as rule based system to test the hypothesis of research in this project. The implementation is based on technical reports

[1]http://www.microsoft.com/enterprisesearch/en/us/fast-customer.aspx

of NERA System developed by Shaalan and Raza (2008). The original NERA system was built using FAST ESP platform. Since the Gate Developer IDE is entirely different from FAST ESP platform, therefore the implementation for Rules provided in NERA reports is not useful with in Gate developer platform. Thus the rules are implemented as JAPE grammar for Gate Developer IDE. From this point onwards we will refer to only newly reproduced Rule Based System as "Named Entity Recognition for Arabic" (NERA). The implementation of NERA system is described in following Sections.

4.3.1 NERA as Corpus Pipeline

NERA is implemented as Gate Corpus Pipeline. A Corpus Pipeline is a Gate application which runs over a Corpus containing documents. The Gate developer does not differentiate between Pattern and Verbatim matchers as described in Section 4.2.1 and Section 4.2.2. Thus new rules have been added with in each phase to incorporate Whitelists into NERA rule based system within Gate Developer.

The main components of NERA corpus pipeline are as follows:

- Arabic Tokenizer
- Gazetteers
- Grammar Rules

The above mentioned components of NERA are described in following sections.

4.3.2 Arabic Tokenizer

The Tokenizer is a processing resource used to identify tokens and their types within the target text. We use built-in Arabic Tokenizer provided with Gate Developer.

4.3.3 Gazetteers

Gazetteers lists built for Named Entity Recognition (described in Chapter 3) are added as ANNIE Gazetteers in Gate Developer; here ANNIE stands for "A Nearly New Information Extraction Systems".

4.3.4 Grammar Rules

Implementing regular expression based rules in Gate Developer requires expertise in Java Annotation Patterns Engine (JAPE). The official documentation for JAPE is available at `http://gate.ac.uk/sale/tao/splitch8.html#chap:jape`. Following are the extracts from the same link:

"JAPE is a Java Annotation Patterns Engine. JAPE provides finite state transduction over annotations based on regular expressions. JAPE is a version of CPSL Common Pattern Specification Language ... 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 consist 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."

The rules for Person, Organization and Location Named Entities were extracted from technical reports of old NERA system prepared by Shaalan and Raza (2008) and some of the rules were enhanced for performance or accuracy. Implementation of JAPE based grammar rules is explained in subsequent sections.

4.3.4.1 Example Rule for Person Names Extractor

Consider the rule (written in regular expression notations) for Person Name as shown in Listing 4.1:

Listing 4.1: Example Person Rule

```
(First Name + (Middle Name)* + Last Name)
```

The rules states that words sequence in target text is annotated as Person Name, if

- The first word is found in First Names Gazetteer for Person Names.
- The middle words (0 or more) are optionally found in Middle Name Gazetteer.
- The last word is found in Last Name Gazetteer for Person Name.

The implementation of the rule as JAPE grammar rule is straightforward in this case as shown in Listing 4.2. The left hand side of the rule is text before the arrow. The ":Per" 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 ":Per" label sequence is given annotation of "Person" and "person". Notice that Kleene star for Middle Name is changed to notation [0,4] which means the JAPE Engine will search for only 0 to 4 instead of 0 to all possible words. This will improve the performance of JAPE rule processing engine significantly.

Listing 4.2: Example Person Rule implemented as JAPE Rule

```
Rule: PersonRule1
Priority:10
(
{Lookup.majorType=="Firsts_v"}
//First Lookup in First name Gazetteer
({Lookup.majorType=="Middle_vv"})[0,4]
//Middle 0 to 4 lookups in Middle name Gazetteer
({Lookup.majorType=="Lasts_v"})?
//Last lookup optionally present in Last name Gazetteer
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="↩
   PersonRule1"}
```

The complete implementation of Person Names Extractor is listed in Appendix A, Section A.1.

4.3.4.2 Example Rule for Organization Names Extractor

Some rules in Organization Name Extractor are based on Derived Adjective Locations (DAL) words. The rule for DAL words are described in Listing 4.3. The implementation of DAL rule as JAPE grammar is not straightforward and required JAVA code in the right hand side of the JAPE rule. The implementation for DAL is shown in Listing 4.4.

Listing 4.3: Derived Adjective Location

```
(ال)?+Country Name + (ي|ية)
```

Listing 4.4: Derived Adjective Location implementation as JAPE rule

```
Rule: DAL01
Priority:10
(
{Lookup.majorType==”Country”}
):ContryForDal
-->
{
 try {
gate.AnnotationSet annSet = (gate.AnnotationSet)bindings.get↩
   (”ContryForDal”);
gate.FeatureMap features = Factory.newFeatureMap();
int start = annSet.firstNode().getOffset().intValue();
int end = annSet.lastNode().getOffset().intValue();
features.put(”initrule”, start+””);
features.put(”endrule”, end+””);
if(start > 0)
 start--;
String content =doc.getContent().toString().substring(start,↩
   start+1);
String aftercontent = ” ”;
try
```

```
{   aftercontent =doc.getContent().toString().substring(end,↩
   end+1);     }
catch(Exception ex) {}
if ( content.equals("ل") )
{
    if(start > 0)
     start--;
    content =doc.getContent().toString().substring(start,↩
       start+1);
    if ( content.equals("ا") )
    {
        if (aftercontent.equals("ى") || aftercontent.equals(↩
           "ي"))
        {
            end++;
            aftercontent =doc.getContent().toString().↩
               substring(end,end+1);
            if (aftercontent.equals("ة")    )
            {
                features.put("rule", "DALRule1");
                outputAS.add((long)start, (long)(end+1), "↩
                   DAL",features);
            }
            else if(aftercontent.equals(" ") || aftercontent↩
               .equals("."))
            {
                features.put("rule", "DALRule1");
                outputAS.add((long)start, (long)(end), "DAL"↩
                   ,features);
            }
        }
    }
}
else  if (aftercontent.equals("ى") || aftercontent.equals("↩
   ي"))
```

```
{
    start++;
    end++;
    aftercontent =doc.getContent().toString().substring(end,↩
       end+1);
    if (aftercontent.equals("ð")    )
    {
        features.put("rule", "DALRule1");
        outputAS.add((long)start, (long)(end+1), "DAL",↩
           features);
    }
    else if(aftercontent.equals(" ") || aftercontent.equals(↩
       "."))
    {
        features.put("rule", "DALRule1");
        outputAS.add((long)start,(long)( end), "DAL",↩
           features);
    }
}
} catch (InvalidOffsetException e) {  throw new ↩
   LuckyException(e); }
}
```

Consider the rule for organization shown in Listing 4.5. The rule states that sequence of words in target text are annotated as Organization if first word is in company preceding know part gazetteer, preceded by optional prefix character and followed by either a DAL or Location Name. The implementation for the rule as JAPE grammar rule is given in Listing 4.6. Notice that rule is split in to two rules for simplicity.

Listing 4.5: Example Organization Rule

```
(ب | ل | ال | لل | و)? company_preceding_known_part + ws + (↩
   DAL|LocationName)
```

Listing 4.6: Example Organization Rule implementation as JAPE rule

```
Rule: Organization1a
Priority:30
(
{Lookup.majorType==”company_preceding_known_part”}
//Word is present in company preceding know part Gazetteers
{DAL} //followed by DAL
):Org
-->
:Org.Organization = {rule=”OrganizationRule01a”}, :Org.↩
   organization = {rule=”OrganizationRule01a”}

Rule: Organization1b
Priority:20
(
{Lookup.majorType==”company_preceding_known_part”}
//Word is present in company preceding know part Gazetteers
{Location}//followed by Location
):Org
-->
:Org.Organization = {rule=”OrganizationRule01b”}, :Org.↩
   organization = {rule=”OrganizationRule01b”}
```

The complete implementation of Organization Names Extractor is listed in Appendix A, Section A.2.

4.3.4.3 Example Rule for Location Names Extractor

Listing 4.7: Example Organization Rule

```
( (عاصمتها) + Any capital name )
```

Consider the rule (written in regular expression notations) for Location Name shown in Listing 4.7:

The rule states that words sequence in target text is annotated as Location Name, if it is preceded by the word عاصمتها. Note that the word may or may not be existing in Gazetteers for Location Name Extractors. The Implementation of the rule as JAPE grammar rule is shown in Listing 4.8.

Listing 4.8: Example Location Rule implemented as JAPE rule

```
Rule: LocationRule4
Priority:12
(
{Token.string=="عاصمتها"}
//If Token matches
({Token})
//then next token must be capital
:AnyCapital
):LOCC
-->
:AnyCapital.Location= {rule="LocationRule4"},:AnyCapital.↩
   location= {rule="LocationRule4"}
```

The complete implementation of Location Names Extractor is listed in Appendix A, Section A.3.

4.3.5 Incorporating Whitelist Mechanism as JAPE Grammar rules in NERA

As mentioned in Section 4.3.1 the Whitelists are added as normal Gazetteer lists in Processing Resources. The verbatim matcher mechanism of old NERA system for Whitelists is incorporated in NERA system by adding JAPE grammar rules. The JAPE rule for Person Name Whitelist is illustrated in Listing 4.9. The Whitelists for Location and Organization Extractors are incorporated in NERA system in the same manner. The same mechanism is applicable for Blacklist for rejecting annotations.

Listing 4.9: Incorporating Whitelist as JAPE rules in NERA

```
Rule: PersonRule9
Priority:10
(
{Lookup.majorType == "Complete_Name"}
//if Lookup found in Complete Name Gazetteer
)
:Per
-->
:Per.Person= {rule="PersonRule9"},:Per.person = {rule="↩
    PersonRule9"}
```

4.3.6 Runtime Parameter Settings for NERA

Following Runtime Parameter setting is required for the resources built for NERA system. The annotationSetName, inputASName and outputASName are all set to "NE" for all processing resources. For PersonGazetteer, longestMatchOnly is set to false and wholeWordsOnly property is set to ture. For LocationGazetteer and OrganizationGazetteer, the longestMatchOnly is set to true and wholeWordsOnly property is set to false. Clicking "Run this Application" will annotate the selected document with named entities which can be saved in XML format.

4.3.7 Integrating NERA with Web Based Interface

The web based system is designed to allow distant/remote researchers and users to use NERA system for preprocessing their documents with Name Entities. In order to integrate NERA system with Web based interface, GATE Embedded framework[1] is used. The example for batch processing available at `http://gate.ac.uk/wiki/code-repository/` is modified and compiled as Java Archive (JAR) file and is used as library in Java Server Pages (JSP) based application. The code for Integrating NERA with Web based application is listed in Appendix B. The Figure 4.3 shows the highlighting of Named Entities in web based interface. The web based system also provides feature for downloading named entities in

[1] http://gate.ac.uk/family/embedded.html

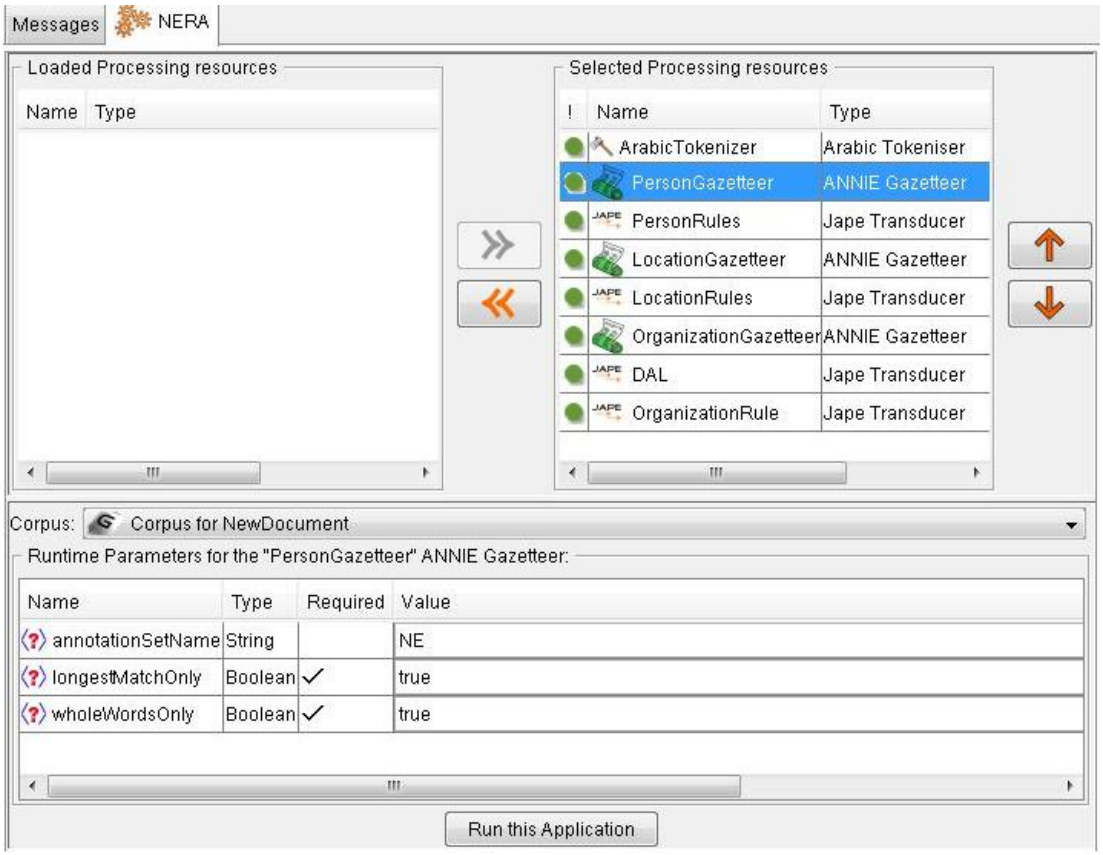

Figure 4.2: Putting it all together - Processing Resources

Figure 4.3: Web Based Interface - Annotations Highlighted in Colors

XML format thus allowing remote users to use Name Entity information in their applications.

4.4 Chapter Summary

This chapter gives overview of NERA system developed by Shaalan and Raza (2008). The chapter also provides overview of Gate Developer IDE along with NERA system implementation in detail using Gate Developer from scratch. The embedding of NERA system with web based application is also described at the end of the chapter. The next chapter is devoted to implementation of Machine Learning System for Named Entity Recognition and integration of rule based system with Machine Learning System.

Chapter 5

Implementation of Machine Learning System

This Chapter describes the Architecture and implementation of Statistical Machine Learning Based System referred to as NERA 2.0 along with the description of integrating Machine Learning and Rule based systems. Furthermore the overview of development/testing Software, "WEKA"[1] is also described briefly.

5.1 Overview and integration of WEKA

WEKA is a 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.

5.2 Feature Set for Machine Learning

Selecting the right Feature set for any Machine Learning application is a crucial task. The large number of features may reduce the performance of learning and

[1] available at `http://www.cs.waikato.ac.nz/ml/weka/`

prediction while cutting down features to minimal number may not be effective for particular purpose. We investigated number of features and selected following features for NER task.

1. The Named Entity tags from NERA system are used as features. Five word sliding window is used for each word in corpus. Thus for every word its own tag along with the tag for two left neighbors and two right neighbors are used. NMinusTwo represent Named Entity tag assigned to the second left neighbor of current word by rule based system. NMinusOne represent tag for immediate left neighbor of current word. N represent tag for current word itself. Similarly NPlusOne and NPlusTwo represent tags for immediate right neighbor and second right neighbor of current word. Sample dataset depicting rule based features for the sentence "... الرئيس الروسي فلاديمير بوتين ..." (Transliterated as Alrais alroosi Flademier Botain) are given in Table 5.1.

2. A boolean feature which is TRUE if the word length is greater then three and FALSE otherwise. As pointed out by Ekbal and Bandyopadhyay (2009) that very small words are rarely Named Entities.

3. A boolean feature which is TRUE if the part of speech Tag is Noun and FALSE otherwise.

4. Part of speech tag for the current word.

5. A boolean feature which is TRUE if the current word is present in Person Gazetteer and FALSE otherwise.

6. A boolean feature which is TRUE if the current word is present in Organization Gazetteer and FALSE otherwise.

7. A boolean feature which is TRUE if the current word is present in Location Gazetteer and FALSE otherwise.

8. A boolean feature which is TRUE if the left neighbor of current word is present in Person Gazetteer and FALSE otherwise.

9. A boolean feature which is TRUE if the left neighbor of current word is present in Organization Gazetteer and FALSE otherwise.
10. A boolean feature which is TRUE if the left neighbor of current word is present in Location Gazetteer and FALSE otherwise.
11. A boolean feature which is TRUE if the right neighbor of current word is present in Person Gazetteer and FALSE otherwise.
12. A boolean feature which is TRUE if the right neighbor of current word is present in Organization Gazetteer and FALSE otherwise.
13. A boolean feature which is TRUE if the right neighbor of current word is present in Location Gazetteer and FALSE otherwise.
14. A feature whose value is "LeftDot" if the left neighbor of current word is full stop '.'. The value is "RightDot" if the right neighbour of current word is full stop '.'. The value is "NONE" otherwise.
15. Prefix of length one for current word
16. Suffix of length one for current word
17. Prefix of length two for current word
18. Suffix of length two for current word
19. Actual Class for training and cross validation

Word	**NMinusTwo**	**NMinusOne**	**N**	**NPlusOne**	**NPlusTwo**	**...**
الرئيس	OTHER	OTHER	OTHER	OTHER	Person	...
الروسي	OTHER	OTHER	OTHER	Person	Person	...
فلاديمير	OTHER	OTHER	Person	Person	OTHER	...
بوتين	OTHER	Person	Person	OTHER	OTHER	...

Table 5.1: Sample Rule based features for 5 Word Window

5.3 Machine Learning Application Architecture

Machine Learning utilizes the features described in Section 5.2. We use supervised learning for enhancing performance of the rule-based Arabic Named Entities recognized by NERA. The architecture of training and testing phases of Machine Learning are discussed in following sections.

5.3.1 Training Phase

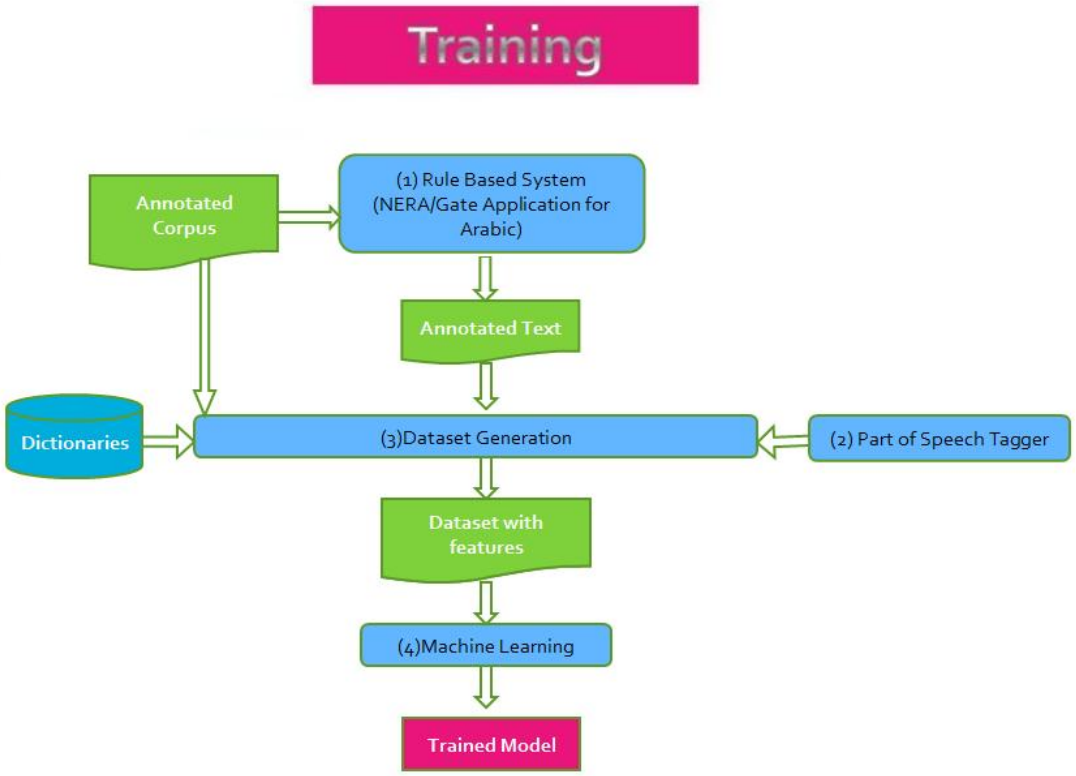

Figure 5.1: Machine Learning Training Phase - Architecture

The flow of Training phase for Machine Learning is shown in Figure 5.1. 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.

5.3.1.1 Application of Rule Based System

The corpora transformed in XML format as described in Section 3.2.3 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.

5.3.1.2 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 JAXP [1]. The java program utilizes Stanford POS Tagger[2] 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 and Organization. Program finally produce features in comma separated values(CSV) format which can be utilized by WEKA Software. The code for dataset generation is Listed in Appendix C.

5.3.1.3 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.

5.3.2 Prediction Phase

The flow of Prediction phase for Machine Learning is shown in Figure 5.2. Though annotated data can be used as 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.

[1] available at https://jaxp.dev.java.net/

[2] available at http://nlp.stanford.edu/software/stanford-postagger-2010-05-26.tgz

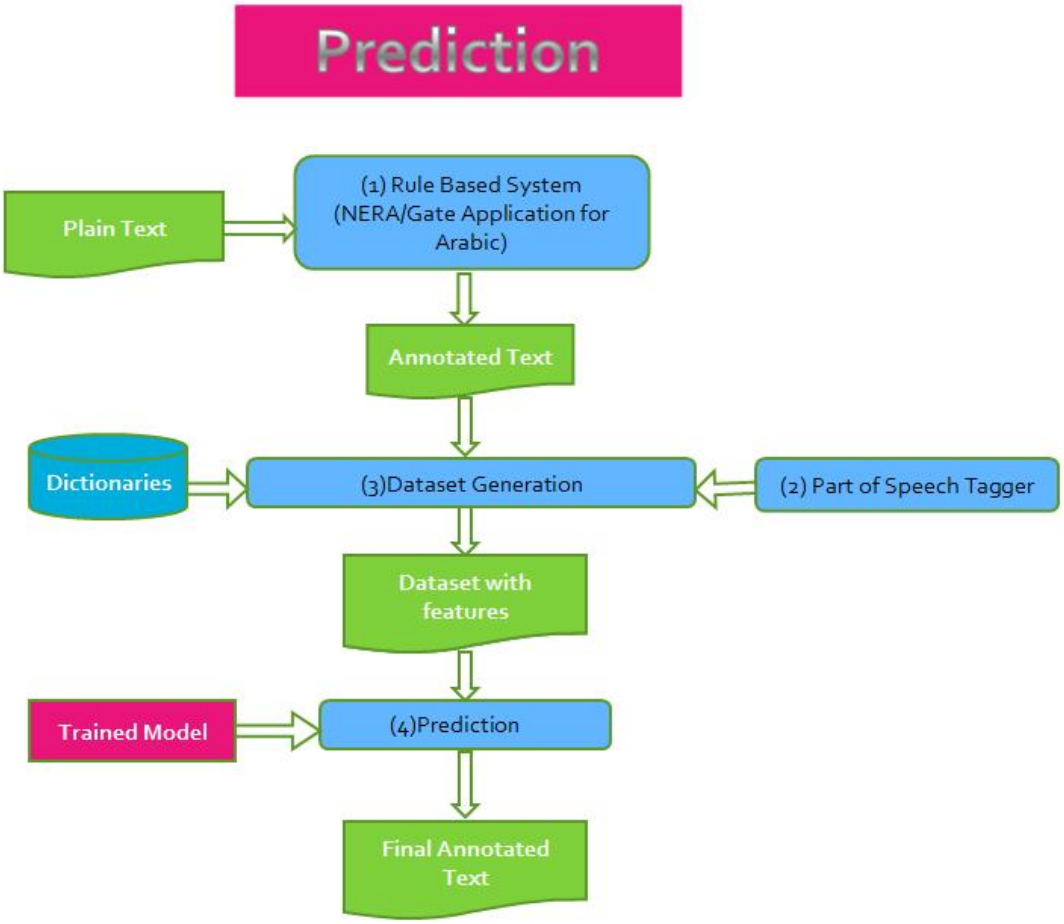

Figure 5.2: Machine Learning Prediction Phase - Architecture

5.3.2.1 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.

5.3.2.2 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.

5.3.2.3 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.

5.4 Chapter Summary

This chapter describes the implementation of Machine Learning System for Named Entity Recognition along with the integration of rule based system with Machine Learning System. Overview of WEKA Software which supports various Machine Learning Tasks is also given. The Chapter also describes the features used for Machine Learning. The next chapter discusses the experimental setup along with the discussion of results achieved.

Chapter 6

Experiments and Results

This chapter elaborates the experiments performed to test the hypothesis of the research along with the discussion of achieved results. The chapter also describes the statistical significance of the results achieved.

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 (De Sitter et al., 2004). The Precision and Recall are given as:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Another way of expressing precision and recall is as follows:

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

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

F-Measue is a harmonic mean that weights Precision and Recall equally and is given as:

$$F - Measure = \frac{2 \times Precision \times Recall}{Precsion + Recall}$$

6.2 Experimental Setup

Experimental Setup is described in following sections:

6.2.1 Dataset Generation

For experiments the datasets are generated using Java code as illustrated in Section 5.3.1.2 using Corpora annotated with NERA System. Three datasets are generated one for each of ANERcorp data, ACE Newswire data and, ACE Boradcast news data.

6.2.2 Classifier Used

WEKA provides several different classifiers that can be applied to data set. We used the following three classifiers for Machine Learning:

1. **J48**, an implementation of C4.5 Algorithm for decision trees by Quinlan (1993).

2. **END**, Ensemble of nested dichotomies for multi-class problems (Dong et al., 2005; Frank and Kramer, 2004).

3. **Bagging** classifier (Breiman, 1996).

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 fold (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 ten fold 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 5.2. The last feature "Actual" represent the target attribute for classification. Each classifier 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 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 ANERcorp corpus and ACE 2003 corpus (Newswire and Broadcast news data) are used for evaluation of systems. These corpora are described in Section 3.2.

The results including Precision (P), recall (R) and F-Measure (F) for ANERcorp from GERA system, NERA system, and three Machine Learning classifiers (J48, Bagging, END) are listed in Table 6.1. The annotations from GERA system are reported for comparison purpose in first row. The results of NERA are shown in second row and serves as baseline. The results of three Machine Learning classifiers are shown in subsequent rows each with three different settings for feature sets. In first setting each Machine Learning Classifier is applied on all the features except rule based features (ML, ML 1, ML 2). In second setting each classifier is applied on only rule based features (MLR, MLR 2, MLR 3). In third setting each classifier is applied on all the features thus representing Hybrid System (Hybrid, Hybrid 2, Hybrid 3). The rule based features correspond to feature number 1 in Section 5.2. The values for F-Measure of Hybrid systems are highlighted in bold. The maximum mean f-measure is also highlighted in bold in tables below.

Table 6.1: Results for ANERcorp Corpus

	Person			Organization			Location			Mean
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**F**
GERA Rule Based System	30.64	12.36	17.61	36.04	22.32	27.57	20.79	2.65	4.7	16.63
NERA Rule Based System (Baseline)	84.51	90.61	87.45	65.12	89.91	75.53	89.22	67.86	77.09	80.02
J48 without Rule Based Features - ML	78.12	70.98	74.38	74.6	59.71	66.33	83.91	72.9	78.02	72.91
J48 with only Rule Based Features - MLR	85.06	90.33	87.62	71.4	86.54	78.24	89.1	67.9	77.07	80.98
J48 with all the Features - Hybrid	94.9	90.78	**92.8**	86.26	85.99	**86.12**	90.6	84.4	**87.39**	88.77
Bagging without Rule Based Features - ML 2	80.82	73.81	77.16	79.39	59.16	67.8	85.03	73.32	78.74	74.57
Bagging with only Rule Based Features - MLR 2	85.03	90.39	87.63	71.24	86.6	78.17	89.19	67.86	77.08	80.96
Bagging with all the Features - Hybrid 2	95.18	91.14	**93.12**	86.28	88.15	**87.21**	91.26	84.53	**87.76**	**89.36**
END without Rule Based Features - ML 3	80.08	71.9	75.77	77.21	58.07	66.29	85.92	72.86	78.86	73.64

Continued on next page

END with only Rule Based Features - MLR 3	84.72	90.49	87.51	71.4	86.54	78.24	89.22	67.86	77.09	80.95
END with all the Features - Hybrid 3	95.29	89.93	**92.53**	86.34	87.07	**86.7**	91.28	83.65	**87.3**	88.84

As evident from the Table 6.1 the results from each Hybrid system outperforms the NERA system, Machine Learning system with only rule based features (MLR), and of the Machine Learning system without rule based features (ML) in terms of F-Measure. Each of the system outperforms the GERA system. Interestingly performance of all of the three Machine Learning classifiers is very similar. Moreover the performance of rule based system is very similar to the performance of Machine Learning system with only rule based features (MLR).

The results from ANERsys 1.0 developed by Benajiba et al. (2007), ANERsys 2.0 developed by Benajiba and Rosso (2007) and Machine Learning system using conditional random fields developed by Benajiba and Rosso (2008) are listed in Table 6.2. Since Benajiba and Rosso (2007, 2008); Benajiba et al. (2007) also performed experiments on ANERcorp corpus, the results are comparable. Clearly the all the Hybrid systems outperform Benajiba and Rosso (2007); Benajiba et al. (2007) for all three Named Entity types. Moreover overall F-Measure of Hybrid 2 system (i.e. 89.36) outperform the overall average F-Measure of Benajiba and Rosso (2008) (i.e. 76.28) for three named entity types by 13.08 points.

Table 6.2: Results for ANERcorp by Benajiba and Rosso (2007, 2008); Benajiba et al. (2007)

	Person			**Organization**			**Location**			**Mean**
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**F**
ANERsys 1.0	54.21	41.01	46.69	45.16	31.04	36.79	82.17	78.42	80.25	54.58
ANERsys 2.0	56.27	48.56	52.13	47.95	45.02	46.43	91.69	82.23	86.71	61.76

Continued on next page

Conditional Random Fields	80.41	67.42	73.35	84.23	53.94	65.76	93.03	86.67	89.74	76.28

The results for ACE Newswire Data as shown in Table 6.3 and results for ACE Broadcast News Data as shown in Table 6.4 also follow the same pattern and the results from Hybrid system outperforms the NERA system, Machine Learning system with only rule based features (MLR), and of the Machine Learning system without rule based features (ML).

Thus the hypothesis of the research described in Section 1.4 held true since Hybrid system (combination of Machine Learning and rule based systems) clearly outperforms the independent rule based system as well as the independent Machine Learning system. The features elaborated in Section 5.2 is the answer of second research question listed in Section 1.4 since Hybrid system performs best when all of the features are used including rule based systems annotations as features.

Table 6.3: Results for ACE Newswire

	Person			**Organization**			**Location**			**Mean**
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**F**
GERA Rule Based System	61.96	58.4	60.13	60.58	53.09	56.59	30.11	3.25	5.86	40.86
NERA Rule Based System (Baseline)	76.85	66.14	71.09	46.6	55.64	50.72	85.05	44.87	58.75	60.19
J48 without Rule Based Features - ML	78.51	72.5	75.38	70.2	37.7	49.05	73.65	55.71	63.44	62.62
J48 with only Rule Based Features - MLR	76.85	66.14	71.09	70.9	46.06	55.84	82.74	47.25	60.15	62.36

Continued on next page

J48 with all the Features - Hybrid	85.62	76.68	**80.9**	78.21	56.12	**65.35**	81.25	61.04	**69.71**	71.99
Bagging without Rule Based Features - ML 2	80.39	72.41	76.19	79.54	37.7	51.15	76.4	55.54	64.32	63.89
Bagging with only Rule Based Features - MLR 2	76.85	66.14	71.09	70.79	45.82	55.63	82.69	47.07	59.99	62.24
Bagging with all the Features - Hybrid 2	88.18	75.98	**81.63**	78.35	55.27	**64.82**	84.93	60.75	**70.83**	**72.43**
END without Rule Based Features - ML 3	80.31	72.06	75.96	77.11	35.52	48.63	78.04	53.57	63.53	62.71
END with only Rule Based Features - MLR 3	76.85	66.14	71.09	69.71	47.15	56.25	83.09	46.72	59.81	62.38
END with all the Features - Hybrid 3	87.6	75.02	**80.83**	83.23	50.55	**62.9**	85.23	59.54	**70.1**	71.28

Table 6.4: Results for ACE Broadcast News

	Person			**Organization**			**Location**			**Mean**
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**F**
GERA Rule Based System	51.24	74.83	60.83	45.43	69.21	54.86	43.02	5.92	10.42	42.04

Continued on next page

NERA Rule Based System (Baseline)	78.41	88.94	83.34	36.8	60.93	45.89	92.88	56.37	70.15	66.46
J48 without Rule Based Features - ML	84.96	78.44	81.57	60.71	33.77	43.4	79.41	67.33	72.88	65.95
J48 with only Rule Based Features - MLR	78.33	88.94	83.3	49.49	32.12	38.96	89.13	59.09	71.06	64.44
J48 with all the Features - Hybrid	91.03	84.76	**87.78**	71	47.02	**56.57**	88.29	72.46	**79.6**	74.65
Bagging without Rule Based Features - ML 2	83.29	79.91	81.57	72.22	30.13	42.52	80.23	73.1	76.5	66.86
Bagging with only Rule Based Features - MLR 2	78.34	88.6	83.16	47.9	26.49	34.12	89.05	59.25	71.15	62.81
Bagging with all the Features - Hybrid 2	91.46	87.02	**89.18**	65.71	45.7	**53.91**	87.96	74.86	**80.88**	**74.66**
END without Rule Based Features - ML 3	83.96	77.99	80.87	67.42	19.87	30.69	80.05	70.7	75.09	62.22
END with only Rule Based Features - MLR 3	78.41	88.94	83.34	48.17	34.77	40.38	91.12	57.49	70.5	64.74

Continued on next page

END with all the Features - Hybrid 3	89.52	85.78	**87.61**	67.84	44.7	**53.89**	89.77	70.94	**79.25**	73.58

6.4 Statistical Significance of Results

The "t-test" is statistical test used to compare results obtained from different groups(human test subjects), algorithms or approaches. The t-test can be applied to check whether the improvement of one approach over another is just by chance or not. The idea is to formulate null hypothesis and alternate hypothesis. The null hypothesis assumes that results of both the approaches are same while the alternate hypothesis represent that results of two approaches are not same. Finally statistical analysis is performed to show that null hypothesis can be rejected thus showing the dominance of one approach over another.

We applied two tailed paired t - test in order to justify that results obtained by our experimentation for datasets are statistically significant and not occurred by chance. This is achieved by showing that F-Measure for Hybrid system is greater then F-Measure for NERA rules based system for each split of copora in ten fold cross validation. Further it is also shown that Machine Learning System with only rule based feature (MLR) and Machine Learning System with other features except rule based features (ML) does not produce desirable results. Thus only Hybrid approach produces improvement over rule based system. The statistical significance experiments are performed only on results produced by J48 Decision Tree classifier. Since the performance of all three Machine Learning classifiers is very similar, these results can be generalized to other two classifiers as well. Reason for choosing J48 Decision Tree Classifier is that it is more robust in performance. Moreover J48 Decision Tree classifier provides the ability to visualize and interpret rules for classification within the Decision Tree.

6.4.1 Statistical Significance of Results for ANERcorp Corpus

In this section the statistical significance of Hybrid approach (using J48 classifier) over NERA rule based system for ANERcorp corpus is evaluated.

Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Machine Learning system without rule based features (ML) is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for ML system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for ML system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for ML system and F-Measure for NERA rule based system (Mean= -6.15, Standard Deviation=4.13, Iterations= 10) is significantly less then zero. Since mean (B-A) is negative, we can conclude that F-Measure of NERA rule based system is greater then F-Measure for ML system. Furthermore the value for t-test (t=-4.71, two tail probability=0.0011) confirms the significance of the results. Thus we can safely reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (-3.2, -9.11). We can thus conclude that NERA rule based system performance is better then ML system.

Now Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Machine Learning system with only rule based features (MLR) is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for MLR system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for MLR system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for MLR system and F-Measure for NERA rule based system (Mean=1.03, Standard Deviation=4.11, Iterations= 10) is close to zero. Furthermore the value for t-test (t=0.79, two tail probability=0.449562) is greater then threshold value (probability $\alpha = 0.05$). Thus we can not reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (-1.91, 3.97). As the confidence interval is small and it includes zero also we can thus accept the null hypothesis. We can conclude that NERA rule based system performance is same as MLR system performance.

Finally Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Hybrid system is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for Hybrid system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for Hybrid system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for Hybrid system and F-Measure for NERA rule based system (Mean=8.68, Standard Deviation=4.1, Iterations=10) is significantly greater then zero. Since mean (B-A) is positive, we can conclude that F-Measure for Hybrid system is greater then F-Measure of NERA rule based system. Furthermore the value for t-test (t=6.696256673, two tail probability=0.000089) confirms the significance of results. Thus we can safely reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (5.75, 11.61). We can conclude that Hybrid system outperforms NERA rule based system.

6.4.2 Statistical Significance of Results for ACE Newswire Corpus

In this section the statistical significance of each Hybrid approach (using J48 classifier) described in Section 6.3 over NERA rule based system for ACE Newswire corpus is evaluated.

Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Machine Learning system without rule based features (ML) is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for ML system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for ML system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for ML system and F-Measure for NERA rule based system (Mean= -3.14, Standard Deviation=3.37, Iterations= 10) is less then zero. Since mean (B-A) is negative, we can conclude that F-Measure of NERA rule based system is greater then F-Measure for ML system. Furthermore the value for t-test (t=-2.95, two tail probability=0.016353) confirms the significance of results. Thus we can safely reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (-5.54, -0.73). We can conclude that NERA rule based system outperforms ML system.

Now Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Machine Learning system with only rule based features (MLR) is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for MLR system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for MLR system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for MLR system and F-Measure for NERA rule based system (Mean=-2.6, Standard Deviation=5.05, Iterations= 10) is close to zero. Furthermore the value for t-test (t=-1.63, two tail probability=0.137472) is greater then threshold value (probability $\alpha = 0.05$). Thus we can not reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (-6.21, 1.01). Although confidence interval includes zero yet it is

not very small, thus we do not accept the null hypothesis either. We conclude that NERA rule based system performance may or may not be the same as MLR system performance.

Finally Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Hybrid system is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for Hybrid system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for Hybrid system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for Hybrid system and F-Measure for NERA rule based system (Mean=15.48, Standard Deviation=14.33, Iterations= 10) is significantly greater then zero. Since mean (B-A) is positive, we can conclude that F-Measure for Hybrid system is greater then F-Measure of NERA rule based system. Furthermore the value for t-test (t=3.42, two tail probability=0.0077) confirms the significance of results. Thus we can safely reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (5.23, 25.73). We can conclude that Hybrid system outperforms NERA rule based system.

6.4.3 Statistical Significance of Results for ACE Broadcast News Corpus

In this section the statistical significance of each Hybrid approach (using J48 classifier) described in Section 6.3 over NERA rule based system for ACE Broadcast News corpus is evaluated.

Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Machine Learning system without rule based features (ML) is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for ML system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for ML system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for ML system and F-Measure for NERA rule based system (Mean= -0.7, Standard Deviation=6.62, Iterations= 10) is close to zero. Furthermore the value for t-test (t=-0.33, two tail probability=0.745468) is greater then threshold value (probability $\alpha = 0.05$). Thus we can not reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (-5.44, 4.04). Although confidence interval includes zero yet it is not very small so we can not accept the null hypothesis either. We conclude that NERA rule based system performance may or may not be the same as ML system performance.

Now Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Machine Learning system with only rule based features (MLR) is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for MLR system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for MLR system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for MLR system and F-Measure for NERA rule based system (Mean=-2.83, Standard Deviation=6.19, Iterations= 10) is close to zero. Furthermore the value for t-test (t=-1.45, two tail probability=0.18167) is greater then threshold value (probability $\alpha = 0.05$). Thus we can not reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (-7.26, 1.59). Although confidence interval includes zero yet it is not very small so we can not accept the null hypothesis either. We conclude that NERA rule based system performance may or may not be the same as MLR system performance.

Finally Let us assume that the F-Measure for NERA rules based system is denoted as "A" and the F-Measure for Hybrid system is denoted as "B". The null hypothesis and alternate hypothesis are described below respectively:

H_0 : The F-Mearuse for Hybrid system is same as F-Measure for NERA rule based system. Thus $B - A = 0$.

H_a : The F-Mearuse for Hybrid system is different from F-Measure for NERA rule based system. Thus $B - A \neq 0$.

The mean difference of F-Measure for Hybrid system and F-Measure for NERA rule based system (Mean=6.55, Standard Deviation=5.59, Iterations= 10) is significantly greater then zero. Since mean (B-A) is positive we can thus conclude that F-Measure for Hybrid system is greater then F-Measure of NERA rule based system. Furthermore the value for t-test (t=3.7, two tail probability=0.0049) confirms the significance of results. Thus we can safely reject the null hypothesis. The 95% Confidence Interval about mean difference in F-Measures is (2.55, 10.55). We can conclude that Hybrid system outperforms NERA rule based system.

6.5 Chapter Summary

In this chapter experiments conducted along with results achieved are discussed. The results confirm the hypothesis of this research. The chapter also describes the evaluation metrics and experimental setup. Ten cross fold validation is used for each Machine Learning classifier. To ensure that results are statistically significant, t-test is applied on the results of J48 classifier as shown in Table 6.5. The next chapter discusses the visualization and application of Decision Trees for J48 Classifier.

Table 6.5: t-Test Results Summary

	Mean Difference	**Standard Deviation**	**t value**	**Two Tail Probability**	**95% Confidence Interval**
ANERcorp Data					
NERA vs ML	-6.15	4.13	-4.71	0.0011	(-3.2,-9.11)

Continued on next page

NERA vs MLR	1.03	4.11	0.79	0.44956	(-1.91,3.97)
NERA vs Hybrid	8.68	4.1	6.70	0.000089	(5.75,11.61)
ACE Newswire Data					
NERA vs ML	-3.14	3.37	-2.95	0.016353	(-5.54,-0.73)
NERA vs MLR	-2.6	5.05	-1.63	0.137472	(-6.2,1.01)
NERA vs Hybrid	15.48	14.33	3.42	0.0077	(5.23,25.73)
ACE Broadcast News Data					
NERA vs ML	-0.7	6.62	-0.33	0.745468	(-5.44,4.04)
NERA vs MLR	-2.83	6.19	-1.45	0.18167	(-7.26,1.59)
NERA vs Hybrid	6.55	5.59	3.7	0.0049	(2.55,10.55)

Chapter 7

Decision Tree Visualization and Application

This chapter describes the visualization and application of Decision Tree for J48 classifier built for dataset generated from ANERcorp corpus. As decision tree can be visualized it is thus possible to interpret the rules within the Decision Tree and analyse path of the Decision Tree used for the classification of particular instances in dataset.

7.1 Decision Tree for J48 Classifier with all the features

WEKA can be used to visualize the Decision Tree model built for J48 classifiers. The tree for J48 classifier with all the features in Section 5.2 applied on ANERcorp Data contains 1126 leaves and the size of the tree is 1684. As Decision Tree is very large, only extract of the Tree for top node N with value Organization is described here. Other extracts of the tree corresponding to top node N with values Person, Location and OTHER are listed in Appendix D. Figrue 7.1 represents the sub tree where top node N has value Location. The extract is interesting as the final class is classify some instances as Location which were identified by rule based system as Organization. The example words which are correctly classified by Model shown in Figure 7.1 are shown in examples below:

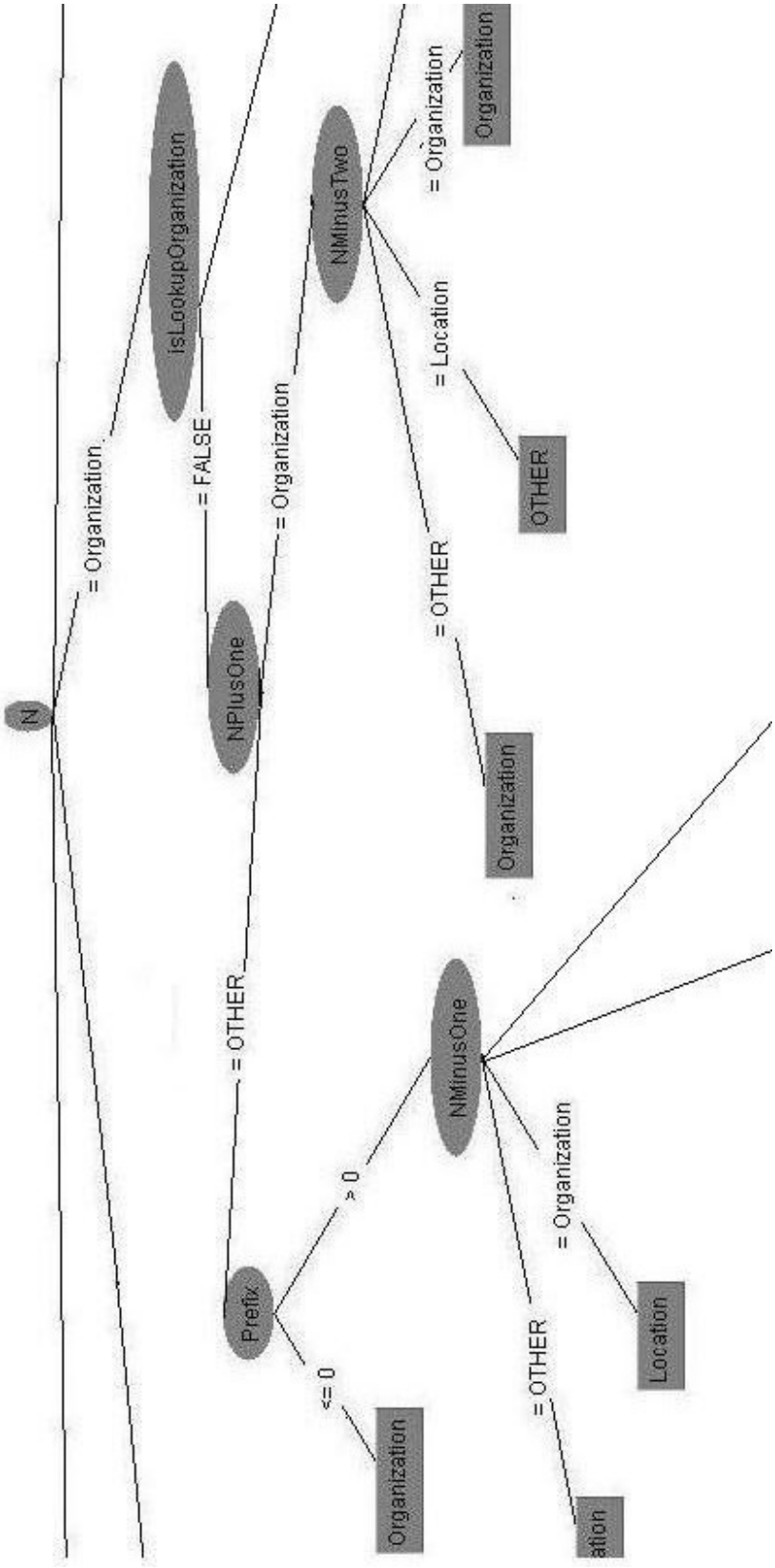

Figure 7.1: Extract of Decision Tree Model - for ANERcorp using all features

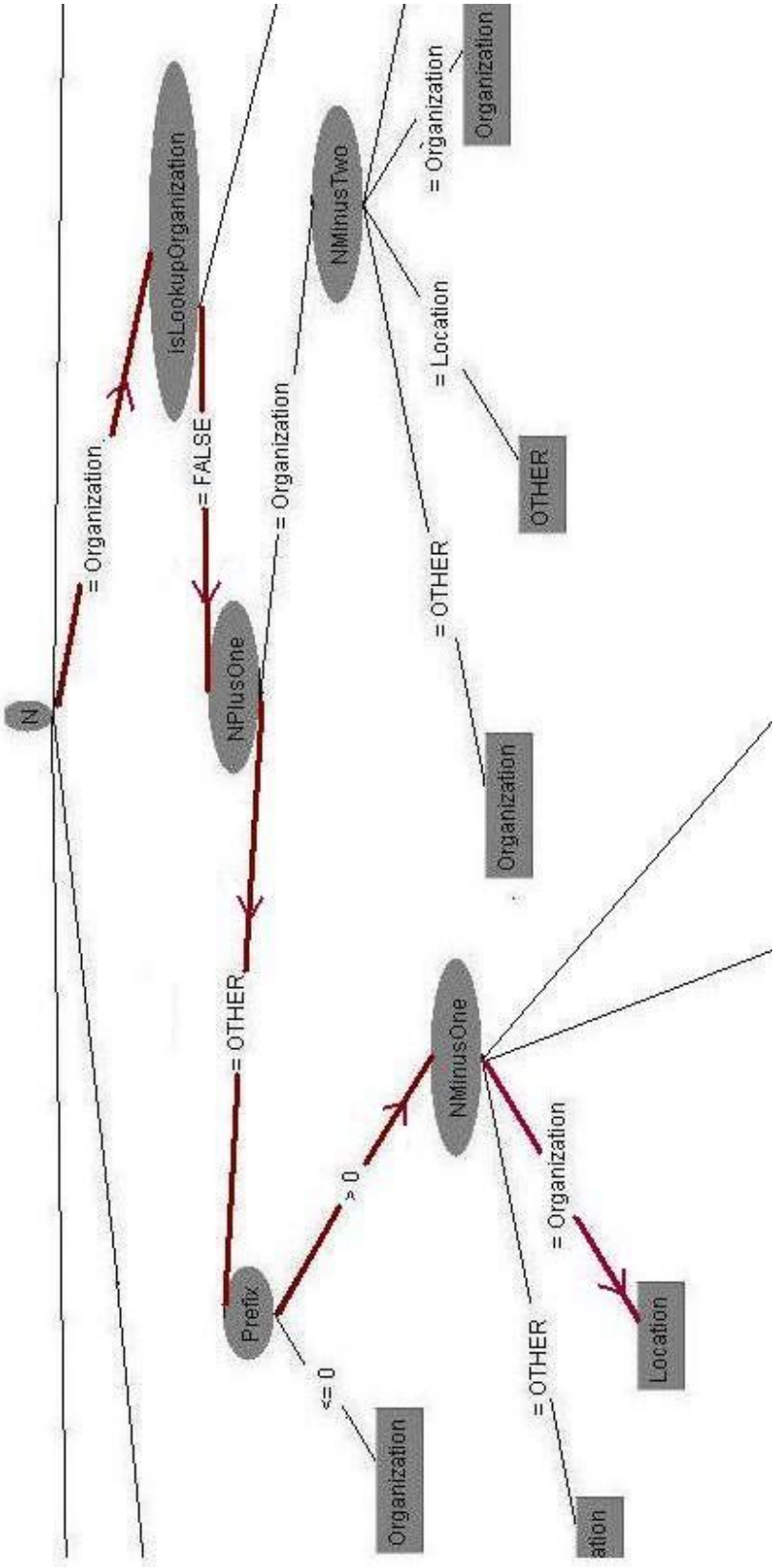

Figure 7.2: Decision Tree Path - for Classification

Consider an example where word ألمانيا is shown with three tags and few words surrounding to the left and right of ألمانيا in the ANERcorp Dataset:

فرانكفورت (د ب أ) أعلن اتحاد صناعة السيارات في
ألمانيا _Location _Location _Organization
امس الاول أن شركات صناعة السيارات في ألمانيا تواجه

In this example the word ألمانيا is followed by first tag "Organization" which is recognised by rule based system. "Location" is the second tag for word ألمانيا and is identified by Decision Tree. The final tag is actual tag in corpus for word ألمانيا and it is also "Location". As per the actual tagging in corpus i.e. "Location", the recognition of word ألمانيا as "Organization" is incorrect by rule based system. The order of tree traversal is given in Figure 7.2 to correctly classify the word as "Location". The values of the features (used in Decision Tree) for this word are N=Organization, isLookupOrganization=FALSE, NPlusOne=OTHER, Prefix=2, NMinusOne=Organization, Actual=Location.

Another similar example is given below:

تحقيق السلام في دارفور . الظواهري قال إن حكومة
الخرطوم _Location _Location _Organization
عاجزة عن حل أزمة دارفور (رويترز) وكان

In this example also the recognition of the word الخرطوم by rule based system as "Organization" is incorrect as it is tagged as "Location" in reference corpus. The correct classification of the word الخرطوم is given by Decision tree as "Location". The the order of tree traversal for is given in Figure 7.2 to correctly classify this word as "Location". The values of features (used in Decision Tree) for this word are N=Organization, isLookupOrganization=FALSE, NPlusOne=OTHER, Prefix=2, NMinusOne=Organization, Actual=Location.

7.2 Decision Tree for J48 Classifier for only rule based features

The Decision Tree model built using ANERcorp Corpus for J48 classifier with only rule based features (MLR) is described here as its ideal to describe because of its small size. The Tree represents the rules in Decision process for predicting classes of new data. Figure 7.3 represent the left subtree of model. As shown in Figure 7.3, if the class of the word is Location from rule based system, then it is classified as Location and in case of Other from rule based system it is classified as Other. In case of current word being classified as Organization from rule based system the system traverse the tree and based on features values, classify the instance. Figure 7.4 represents the right subtree of model.

7.3 Chapter Summary

In this chapter the Decision Tree Model built for J48 classifier is discussed. The Decision Tree is visualized along with sample rule applied from the Decision Tree on example data. The next chapter discusses the conclusion of the research. The conclusion is drawn based on the achieved results along with the outline for intended future research.

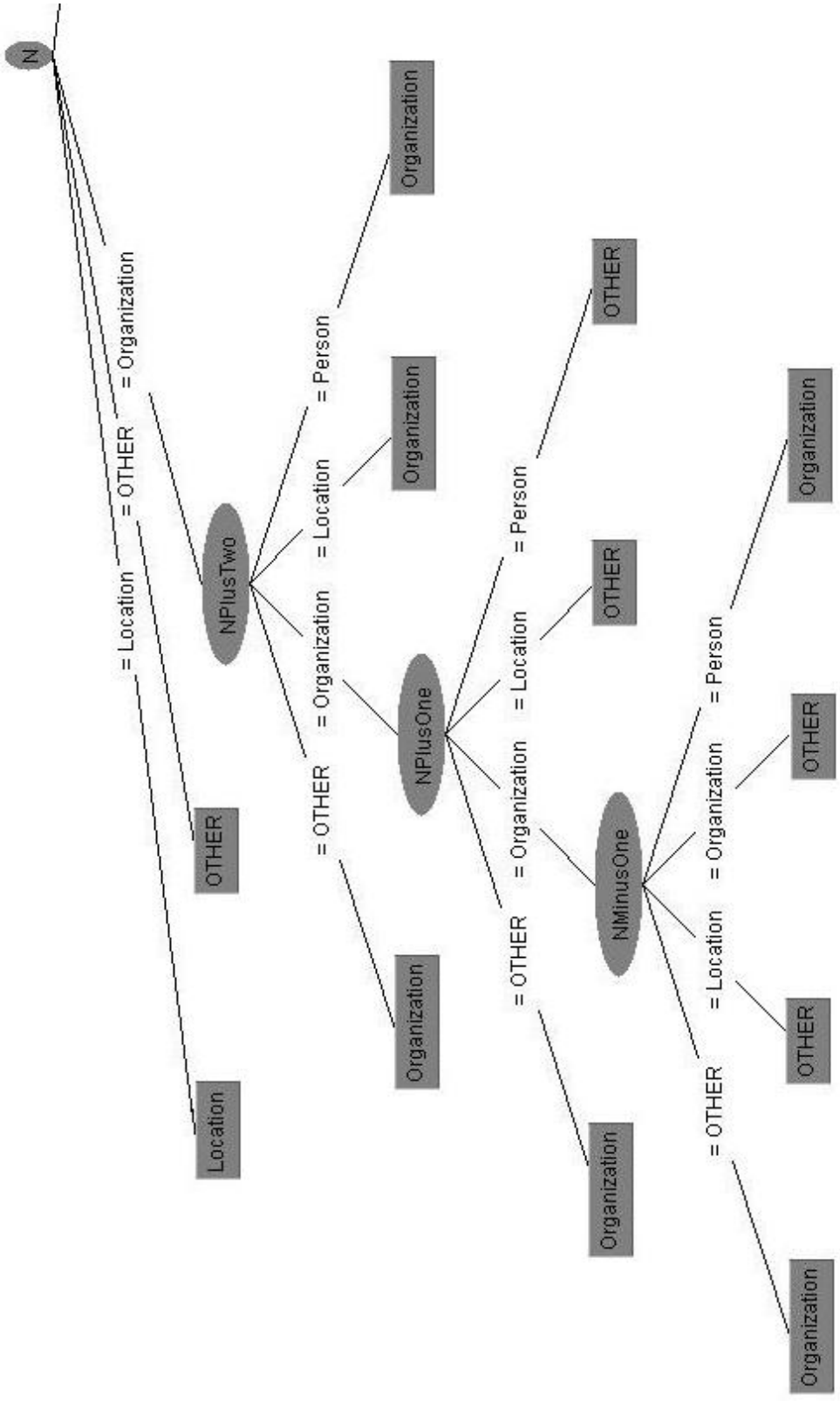

Figure 7.3: Left subtree of Decision Tree - for ANERcorp Corpus using rule based features only

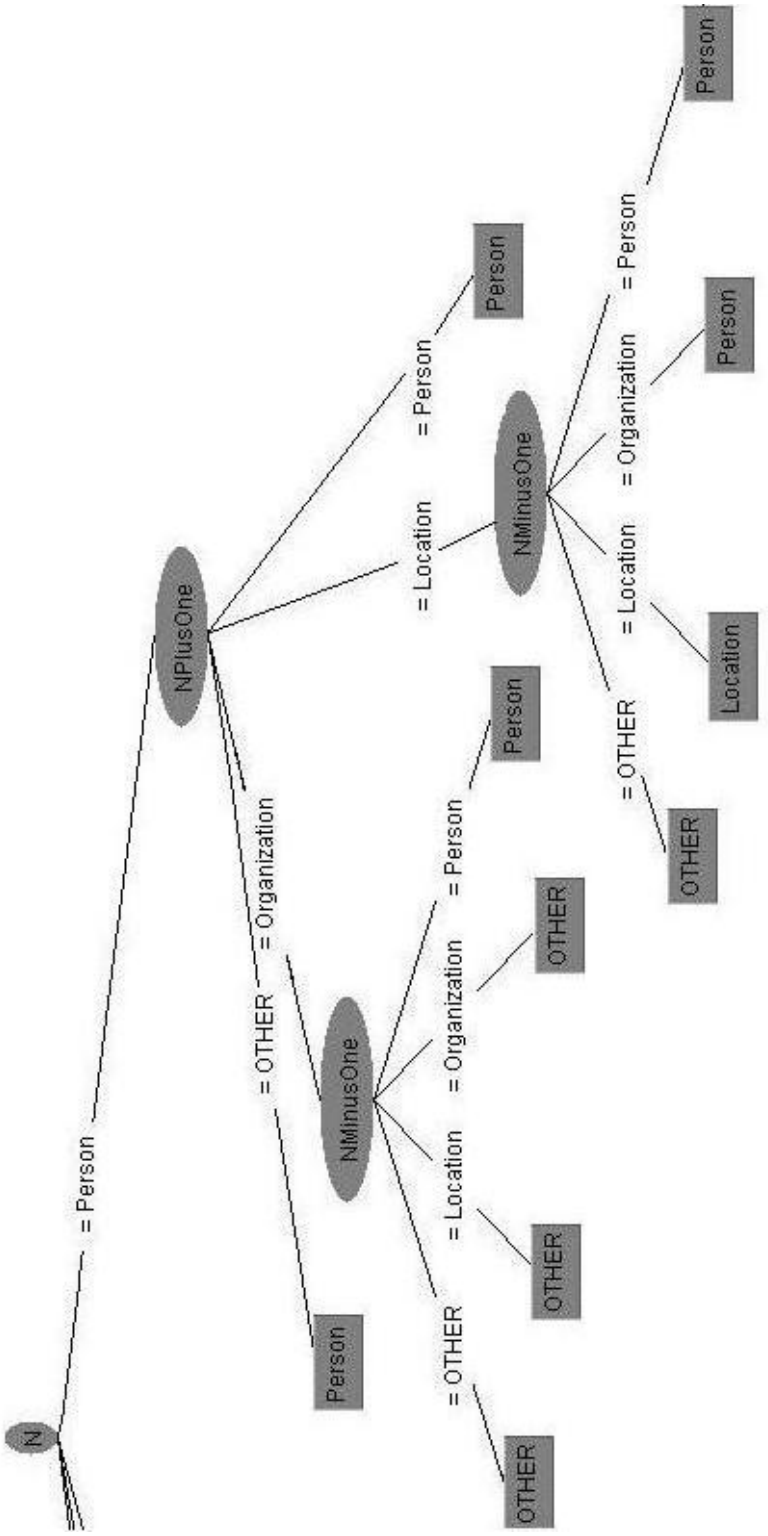

Figure 7.4: Right subtree of Decision Tree - for ANERcorp Corpus using rule based features only

Chapter 8

Conclusion and Future Work

This chapter draws conclusion about the research. The chapter also put light on Future directions for the research.

8.1 Conclusion

This research project is an attempt to improve the rule based Named Entity Recognition System by means of applying Machine Learning. The research is first of its kind as no Hybrid system is found published for Arabic Named Entity Recognition. The thesis describes the Data Collection mechanism and implementation of rule based and Machine Learning systems.

The NERA system that is reproduced as GATE Developer application from scratch. The rules for Person, Location and Organization Name extractors are covered in NERA system. The NERA system is used as rule based system for generating Named Entity annotation for Corpora including ANERcorp corpus, ACE Newswire corpus and ACE Broadcast News corpus. Three datasets are generated using these copora which includes features described in Section 5.2 for each word present in corpora. Machine Learning classifiers J48, Bagging and END are applied on each dataset with and without rule based features (annotations). The classifiers are also applied on data with all the features, which represents Hybrid systems. The ten fold cross validation methodology is used for Machine Learning system evaluation.

The comparisons of experimental results are evident that performance of rule based system can be enhanced by application of Machine Learning system which uses annotations of rule based system as features in conjunction with several other features. The Hybrid approach described in the thesis clearly outperformed independent rule based system and independent Machine Learning system. The ten fold cross validation methodology for Machine Learning system ensures that the results obtained are valid and does not over fit. Application of three different classifiers shows the same trend. The three classifiers produce similar results though on average "Bagging" outperformed others with overall average F-Measure 89.36

The statistical significance test "t-test" is also applied on results obtained to check whether the results are statistically significant and not occurred by chance. The t-test results confirm the validity of hypothesis of this research, since the Hybrid system outperformed the NERA rule based system, Machine Learning system with only rule based features and Machine Learning system without rule based features. The answer for the first research question described in Section 1.4 is thus affirmative. Hybrid system utilizes all the features described in Section 5.2. Thus these features are answer to second research question.

Although the experiments are conducted on Arabic language the system can seamlessly be integrated with rule based system targeted at other languages. It is expected that approach will have similar effects of performance improvement on other languages.

The major issue with Arabic Named Entity Recognition is the lack of resources including tagged corpora for Named Entity Recognition and Gazetteers. It is anticipated that large and accurately annotated corpora along with large Gazetteers will result in better model of Machine Learning and Classification.

8.2 Future Directions

The project is a work in progress. Only three Named Entity types including Person, Organization and Location which corresponds to ENAMEX tag of Message Understanding Conference are implemented and investigated. It is expected that increasing the coverage of Named Entities along with large annotated corpus can improve the performance for all the covered entities. The lack of resources is a

major issue faced during the project. Only small number of Gazetteers with limited words are built and more data is to be added to gazetteers. It is planned to improve the quality of current rules along with addition of new rules for the same entities. This requires study of large annotated corpora and linguistic efforts.

Web based system is intended to include features for adding and updating Gazetteer lists. Though web based system supports downloading annotated text, it is intended to build a web service to facilitate NLP researchers. Web service will enable seamless integration of Named Entity Recognition for Arabic system with other Natural Language processing systems and agents thus reducing human efforts.

For future work, it is also intended to increase the number of features for Machine Learning. Moreover only few Classifiers for Machine Learning were tested and results from J48, Bagging and END classifiers are reported. In future more Machine Learning classifiers are planned to be applied and evaluated on datasets.

References

Abdul Hamid, A. and Darwish, K. (2010), Simplified feature set for arabic named entity recognition, *in* 'Proceedings of the 2010 Named Entities Workshop', Association for Computational Linguistics, Uppsala, Sweden, pp. 110–115.
URL: *http://www.aclweb.org/anthology/W10-2417* 12

Alpaydin, E. (2004), *Introduction to Machine Learning*, The MIT Press. 11

Attia, M., Toral, A., Tounsi, L., Monachini, M. and van Genabith, J. (2010), An automatically built named entity lexicon for arabic, *in* N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner and D. Tapias, eds, 'Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)', European Language Resources Association (ELRA), Valletta, Malta. 9

Baluja, S., Mittal, V. O. and Sukthankar, R. (2000), "Applying machine learning for high performance named-entity extraction", *Computational Intelligence* , Vol. 16, pp. 586–595. 11

Benajiba, Y., Diab, M. and Rosso, P. (2008*a*), Arabic named entity recognition: An svm-based approach, *in* 'The International Arab Conference on Information Technology', ACIT2008. 13

Benajiba, Y., Diab, M. and Rosso, P. (2008*b*), Arabic named entity recognition using optimized feature sets, *in* 'Proceedings of the Conference on Empirical Methods in Natural Language Processing', EMNLP '08, Association for Computational Linguistics, Morristown, NJ, USA, pp. 284–293. 2, 6, 13

Benajiba, Y. and Rosso, P. (2007), Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information, *in* 'IICAI', pp. 1814–1823. viii, 8, 12, 22, 58

Benajiba, Y. and Rosso, P. (2008), Arabic named entity recognition using conditional random fields, *in* 'Workshop on HLT & NLP within the Arabic world. Arabic Language and local languages processing: Status Updates and Prospects'. viii, 6, 12, 58

Benajiba, Y., Rosso, P. and Benedí Ruiz, J. M. (2007), Anersys: An arabic named entity recognition system based on maximum entropy, *in* 'Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing', CICLing '07, Springer-Verlag, Berlin, Heidelberg, pp. 143–153. viii, 11, 22, 58

Benajiba, Y., Zitouni, I., Diab, M. and Rosso, P. (2010), Arabic named entity recognition: using features extracted from noisy data, *in* 'Proceedings of the ACL 2010 Conference Short Papers', ACLShort '10, Association for Computational Linguistics, Morristown, NJ, USA, pp. 281–285. 13

Biswas, S., Mishra, S. P., Acharya, S. and Mohanty, S. (2010), "A hybrid oriya named entity recognition system: Harnessing the power of rule", *International Journal of Artificial Intelligence and Expert Systems (IJAE)* , Vol. 1, pp. 1–6. 15

Breiman, L. (1996), "Bagging predictors", *Mach. Learn.* , Vol. 24, Kluwer Academic Publishers, Hingham, MA, USA, pp. 123–140.
URL: *http://portal.acm.org/citation.cfm?id=231986.231989* 55

Cunningham, H., Maynard, D., Bontcheva, K. and Tablan, V. (2002), Gate: A framework and graphical development environment for robust nlp tools and applications, *in* 'Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics'. 19, 33

De Sitter, A., Calders, T. and Daelemans, W. (2004), "A formal framework for evaluation of information extraction", Antwerp.
URL: *http://www.cnts.ua.ac.be/papers/2004/ef04.pdf* 54

Dong, L., Frank, E. and Kramer, S. (2005), Ensembles of balanced nested dichotomies for multi-class problems, *in* 'PKDD', Springer, pp. 84–95. 55

Ekbal, A. and Bandyopadhyay, S. (2009), Voted ner system using appropriate unlabeled data, *in* 'Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration', NEWS '09, Association for Computational Linguistics, Morristown, NJ, USA, pp. 202–210. 12, 13, 48

Ekbal, A. and Bandyopadhyay, S. (2010*a*), "Named entity recognition using appropriate unlabeled data, post-processing and voting", *Informatica* , Vol. 34, pp. 55–76. 13

Ekbal, A. and Bandyopadhyay, S. (2010*b*), "Named entity recognition using support vector machine: A language independent approach", *International Journal of Electrical, Computer, and Systems Engineering* , Vol. 4, pp. 155–170. 13

Elsebai, A., Meziane, F. and Belkredim, F. Z. (2009), A rule based persons names arabic extraction system, *in* 'Proceedings of the IBIMA', 4-6 January, 2009, Cairo, Egypt. 10

Ferreira, E., Balsa, J. and Branco, A. (2007), 'A.: Combining rule-based and statistical methods for named entity recognition in portuguese. in: Actas da 5 a workshop em tecnologias da informao e da linguagem humana'. 15

Frank, E. and Kramer, S. (2004), Ensembles of nested dichotomies for multi-class problems, *in* 'Twenty-first International Conference on Machine Learning', ACM. 55

Habash, N. Y. (2010), *Introduction to Arabic Natural Language Processing*, Morgan & Claypool Publisher. 7

han Tsai, T., Wu, S.-H. and lian Hsu, W. (2003), Mencius: A chinese named entity recognizer using hybrid model, *in* 'Computational Linguistics & Chinese Language Processing (9) 2004', pp. 65–82. 15

Maloney, J. and Niv, M. (1998), Tagarab: a fast, accurate arabic name recognizer using high-precision morphological analysis, *in* 'Proceedings of the Workshop on Computational Approaches to Semitic Languages', Semitic '98, Association for Computational Linguistics, Morristown, NJ, USA, pp. 8–15. 10

Mao, X., Xu, W., Dong, Y., He, S. and Wang, H. (2007), Using non-local features to improve named entity recognition recall, *in* 'Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation', PACLIC 07. 13

Mayfield, J., McNamee, P. and Piatko, C. (2003), Named entity recognition using hundreds of thousands of features, *in* 'Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4', Association for Computational Linguistics, Morristown, NJ, USA, pp. 184–187.
URL: *http://dx.doi.org/10.3115/1119176.1119205* 12

Nguyen, H. T. and Cao, T. H. (2008), Named entity disambiguation: A hybrid statistical and rule-based incremental approach, *in* 'Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web', ASWC '08, Springer-Verlag, Berlin, Heidelberg, pp. 420–433. 14

Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V. and Spyropoulos, C. D. (2001), Using machine learning to maintain rule-based named-entity recognition and classification systems, *in* 'Proceedings of the 39th Annual Meeting on Association for Computational Linguistics', ACL '01, Association for Computational Linguistics, Morristown, NJ, USA, pp. 426–433. 15, 16, 17

Quinlan, J. R. (1993), *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 55

Riaz, K. (2010), Rule-based named entity recognition in urdu, *in* 'Proceedings of the 2010 Named Entities Workshop', ACL 2010, pp. 126–135. 10

Shaalan, K. and Raza, H. (2007), Person name entity recognition for arabic, *in* 'Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources', Semitic '07, Association for Computational Linguistics, Morristown, NJ, USA, pp. 17–24. 10

Shaalan, K. and Raza, H. (2008), Arabic named entity recognition from diverse text types, *in* 'Proceedings of the 6th international conference on Advances in Natural Language Processing', GoTAL '08, Springer-Verlag, Berlin, Heidelberg, pp. 440–451. 1, 2, 7, 8, 10, 18, 34, 35, 36, 37, 46

Shaalan, K. and Raza, H. (2009), "Nera: Named entity recognition for arabic", *Journal of the American Society for Information Science and Technology* , pp. 1652–1663. 7, 8, 10

Srihari, R., Niu, C. and Li, W. (2000), A hybrid approach for named entity and sub-type tagging, *in* 'Proceedings of the sixth conference on Applied natural language processing', Association for Computational Linguistics, Morristown, NJ, USA, pp. 247–254. 14

Traboulsi, H. (2009), Arabic named entity extraction: A local grammar-based approach, *in* 'Proceedings of the International Multiconference on Computer Science and Information Technology', Volume 4, pp. 139–143. 10

Zhang, L., Pan, Y. and Zhang, T. (2004), Focused named entity recognition using machine learning, *in* 'Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval', SIGIR '04, ACM, New York, NY, USA, pp. 281–288. 13, 14

Appendix A

JAPE Implementation for Rules

A.1 Person Name Extractor Rules

Listing A.1: JAPE Rules for Person Name Extractor

```
Phase:nestedpatternphase
Input:  Lookup Token
//note that we are using Lookup and Token both inside our ↩
   rules.
Options: control = appelt Debug=true

Rule: PersonRule9
Priority:10
(
{Lookup.majorType == "Complete_Name"}
)
:Per
-->
:Per.Person= {rule="PersonRule9"},:Per.person = {rule="↩
   PersonRule9"}

Rule: PersonRule8
Priority:17
({Lookup.majorType == "person_title2"}):personTitle
```

```
(
({Lookup.majorType == "Firsts_v"}|{Lookup.majorType == "Middle_vv"}|{Lookup.majorType == "Lasts_v"})
({Lookup.majorType == "Firsts_v"}|{Lookup.majorType == "Middle_vv"}|{Lookup.majorType == "Lasts_v"})[0,5]
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="PersonRule8"}



Rule: PersonRule7
Priority:16
(
{Lookup.majorType == "Firsts_v"}
({Lookup.majorType == "Middle_vv"}|{Lookup.majorType == "Lasts_v"})?
({Lookup.majorType == "Laqab"})+
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="PersonRule7"}


Rule: PersonRule6
Priority:15
(
(
    {Lookup.majorType=="Firsts_v"}
    ({Token.string == "ابن"}|{Token.string == "أبن"}|{Token.string == "بن"}|{Token.string == "أبي"}|{Token.string == "ابي"}|{Token.string == "بنت"})+
```

```
    {Token.string != ""}
)
(
    ({Token.string == "        "}|{Token.string == "أبن"}|{↩
       Token.string == "بن"}|{Token.string == "أبي"}|{Token.↩
       string == "ابي"}|{Token.string == "بنت"})+
    {Token.string !=""}

)?
)
:Per
-->
:Per.Person= {rule="PersonRule6"},:Per.person = {rule="↩
   PersonRule6"}


Rule: PersonRule61
Priority:15
(
(
    {Lookup.majorType!="Firsts_v"}
    ({Token.string == "ابن"}|{Token.string == "أبن"}|{Token.↩
       string == "بن"}|{Token.string == "أبي"}|{Token.string ↩
       == "ابي"}|{Token.string == "بنت"})+
    {Token.string != ""}
)
(
    ({Token.string == "ابن"}|{Token.string == "أبن"}|{Token.↩
       string == "بن"}|{Token.string == "أبي"}|{Token.string ↩
       == "ابي"}|{Token.string == "بنت"})+
    {Token.string !=""}

)?
)
```

```
:Per
-->
:Per.Person= {rule="PersonRule61"},:Per.person = {rule="↩
    PersonRule61"}



Rule: PersonRule5
Priority:14
(
({Token.string == "ابو"}|{Token.string == "ام"})
{Lookup.majorType == "Firsts_v"}
({Lookup.majorType == "Firsts_v"}|{Lookup.majorType == "↩
    Middle_vv"}|{Lookup.majorType == "Lasts_v"})?
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="↩
    PersonRule5"}


Rule: PersonRule51
Priority:14
(
({Token.string == "ابو"}|{Token.string == "ام"})
{Lookup.majorType != "Firsts_v"}
({Lookup.majorType == "Firsts_v"}|{Lookup.majorType == "↩
    Middle_vv"}|{Lookup.majorType == "Lasts_v"})?
)
:Per
-->
:Per.Person= {rule="PersonRule51"},:Per.person = {rule="↩
    PersonRule51"}


Rule: PersonRule4
```

```
Priority:13
(
{Lookup.majorType=="Honor"}
({Lookup.majorType=="Location"})?
):honorandLoc
(
{Lookup.majorType=="Firsts_v"}
({Lookup.majorType=="Middle_vv"})?
({Lookup.majorType=="Lasts_v"})?
({Lookup.majorType=="Number1"})?
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="↩
    PersonRule4"}


Rule: PersonRule3
Priority:12
(
{Lookup.majorType=="Firsts_v"}
({Lookup.majorType=="Middle_vv"})?
{Lookup.majorType=="Lasts_v"}
)
:Per
(
{Lookup.majorType=="Person_indicator"}
):Pre_Per_Ind
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="↩
    PersonRule3"}


Rule: PersonRule2
Priority:11
(
```

```
{Lookup.majorType=="Person_indicator"}
):Pre_Per_Ind
(
{Lookup.majorType=="Firsts_v"}
({Lookup.majorType=="Middle_vv"})?
{Lookup.majorType=="Lasts_v"}
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="↩
    PersonRule2"}

Rule: PersonRule1
Priority:10
(
{Lookup.majorType=="Firsts_v"}
({Lookup.majorType=="Middle_vv"})[0,4]
({Lookup.majorType=="Lasts_v"})?
)
:Per
-->
:Per.Person= {rule="PersonRule1"},:Per.person = {rule="↩
    PersonRule1"}
```

A.2 Organization Name Extractor Rules

Listing A.2: JAPE Rules for Organization Extractor

```
Phase: Organization2
Input:  Lookup DAL Location Token
Options: control = appelt Debug=true

Rule: Organization8
Priority:30
```

```
(
{Lookup.majorType=="company_preceding_known_part"}
{Token.string=="("}
{Token}
{Token.string==")"}
):Org
-->
:Org.Organization = {rule="OrganizationRule8"}, :Org.↩
   organization = {rule="OrganizationRule8"}

Rule: Organization7
Priority:30
(
({Lookup.majorType=="company_preceding_known_part"}|{Lookup.↩
   majorType=="company_preceding_indicator"})
{Token.string=="\""}
({Token})
{Token.string=="\""}
({DAL}
|
{Token.string=="و\""}{Token}{Token.string=="\""}
)
):Org
-->
:Org.Organization = {rule="OrganizationRule7"}, :Org.↩
   organization = {rule="OrganizationRule7"}

Rule: Organization6
Priority:30
(
({Lookup.majorType=="company_preceding_known_part"}|{Lookup.↩
   majorType=="company_preceding_indicator"})
    (
    {DAL}
    |
    {Location}
```

```
    |
    {Lookup.majorType=="businessType"}
    )
{Token}
):Org
-->
:Org.Organization = {rule="OrganizationRule6"}, :Org.↩
   organization = {rule="OrganizationRule6"}

Rule: Organization5
Priority:30
(
({Lookup.majorType=="company_preceding_known_part"}|{Lookup.↩
   majorType=="company_preceding_indicator"})
({Token})[0,4]
(
    {DAL}
    |
    {Lookup.majorType=="company_following_indicator"}
    |
    (   ({Lookup.majorType=="prefix_buisness"})?
        {Lookup.majorType=="businessType"}
        ({DAL})?
    )
)
):Org
-->
:Org.Organization = {rule="OrganizationRule5"}, :Org.↩
   organization = {rule="OrganizationRule5"}

Rule: Organization4
Priority:30
(
{Lookup.majorType=="company_preceding_known_part"}
{DAL}
{Token.string=="\""}
```

```
({Token})[0,4]
{Token.string=="\""}
):Org
-->
:Org.Organization = {rule="OrganizationRule4"}, :Org.↩
   organization = {rule="OrganizationRule4"}

Rule: Organization3
Priority:30
(
{Lookup.majorType=="company_preceding_known_part"}
({Token})[1,4]
({Token.string=="في"})?
{Location}

):Org
-->
:Org.Organization = {rule="OrganizationRule3"}, :Org.↩
   organization = {rule="OrganizationRule3"}

Rule: Organization2
Priority:30
(
{Lookup.majorType=="company_preceding_known_part"}
({Token})[0,4]
(
({Lookup.majorType=="company_following_known_part"})
|
({DAL}
{Token.string=="للتلفزيون"})
|
({Token.string=="التلفزيونية"}{DAL})
)
):Org
-->
```

```
:Org.Organization = {rule="OrganizationRule2"}, :Org.↩
   organization = {rule="OrganizationRule2"}


Rule: Organization1a
Priority:30
(
{Lookup.majorType=="company_preceding_known_part"}
{DAL}
):Org
-->
:Org.Organization = {rule="OrganizationRule01a"}, :Org.↩
   organization = {rule="OrganizationRule01a"}

Rule: Organization1b
Priority:20
(
{Lookup.majorType=="company_preceding_known_part"}
{Location}
):Org
-->
:Org.Organization = {rule="OrganizationRule01b"}, :Org.↩
   organization = {rule="OrganizationRule01b"}


Rule: OrganizationRule01
Priority:10
(
{Lookup.majorType=="Org"}
):Org
-->
:Org.Organization = {rule="OrganizationRule0"}, :Org.↩
   organization = {rule="OrganizationRule0"}

Rule: OrganizationRule00
Priority:10
```

```
(
{Lookup.majorType=="Organizations"}
):Org
-->
:Org.Organization = {rule="OrganizationRule00"}, :Org.↩
   organization = {rule="OrganizationRule00"}
```

A.3 Location Name Extractor Rules

Listing A.3: JAPE Rules for Location Extractor

```
Phase:Location4
Input:  Lookup Token
Options: control = brill Debug=true

Rule: LocationRule19
Priority:19
(
{Token.string=="الي"}
{Lookup.majorType=="Direction2"}

{Token.string=="من"}
({Lookup.majorType=="Locations"}
):Loc
):LOCC
-->
:Loc.Location= {rule="LocationRule19"},:Loc.location= {rule=↩
   "LocationRule19"}

Rule: LocationRule18
Priority:19
(
({Lookup.majorType=="City"})
({Token.string=="("})?
```

```
({Lookup.majorType=="Country"}
|{Lookup.majorType=="State"}):Loc
({Token.string==")"})?
):LOCC
-->
:Loc.Location= {rule="LocationRule18"},:Loc.location= {rule=↩
   "LocationRule18"}

Rule: LocationRule17
Priority:19
(
({Lookup.majorType=="City"}):Loc
({Token.string=="("})?
({Lookup.majorType=="Country"}
|{Lookup.majorType=="State"})
({Token.string==")"})?
):LOCC
-->
:Loc.Location= {rule="LocationRule17"},:Loc.location= {rule=↩
   "LocationRule17"}

Rule: LocationRule16
Priority:19
(
({!Lookup}):Loc
{Token.string=="و"}
```

```
({Lookup.majorType=="Country"}
|{Lookup.majorType=="City"}|{Lookup.majorType=="State"}|{↩
   Lookup.majorType=="Capital"})
):LOCC
-->
:Loc.Location= {rule="LocationRule16"},:Loc.location= {rule=↩
   "LocationRule16"}

Rule: LocationRule15
Priority:19
```

```
(
({Lookup.majorType=="Country"}
|{Lookup.majorType=="City"}|{Lookup.majorType=="State"}|{↩
   Lookup.majorType=="Capital"})
{Token.string=="و"}
```

```
({!Lookup}):Loc
):LOCC
-->
:Loc.Location= {rule="LocationRule15"},:Loc.location= {rule=↩
   "LocationRule15"}

Rule: LocationRule14
Priority:19
(
({Lookup.majorType=="Direction2"}|{Lookup.majorType=="↩
   Direction4"}|{Lookup.majorType=="Direction5"})
{Token.string=="من"}
```

```
(
{Lookup.majorType=="Country"}
|{Lookup.majorType=="City"}|{Lookup.majorType=="State"}|{↩
   Lookup.majorType=="Capital"}):Loc
):LOCC
-->
:Loc.Location= {rule="LocationRule14"},:Loc.location= {rule=↩
   "LocationRule14"}

Rule: LocationRule13
Priority:18
(
{Token.string=="دول"}
```

```
({Lookup.majorType=="Direction1"}|{Lookup.majorType=="↩
   Direction3"})?
({Lookup.majorType=="Continents"}):Loc
):LOCC
-->
```

```
:Loc.Location= {rule="LocationRule13"},:Loc.location= {rule=↩
    "LocationRule13"}

Rule: LocationRule12
Priority:17
(
({Lookup.majorType=="AdmDiv"}|{Token.string=="مدينة"})
({Lookup.majorType=="City"}):Loc
({Lookup.majorType=="Direction1"}|{Lookup.majorType=="↩
    Direction2"}|{Lookup.majorType=="Direction3"}|{Lookup.↩
    majorType=="Direction4"})
):LOCC
-->
:Loc.Location= {rule="LocationRule12"},:Loc.location= {rule=↩
    "LocationRule12"}


Rule: LocationRule11
Priority:16
(
({Lookup.majorType=="City"}):Loc
{Token.string=="في"}
({Lookup.majorType=="Country"}|{Lookup.majorType=="Capital"↩
    }|{Lookup.majorType=="State"})
):LOCC
-->
:Loc.Location= {rule="LocationRule11"},:Loc.location= {rule=↩
    "LocationRule11"}

Rule: LocationRule10
Priority:16
(
{Lookup.majorType=="City"}
{Token.string=="في"}
({Lookup.majorType=="Country"}|{Lookup.majorType=="Capital"↩
    }|{Lookup.majorType=="State"}):Loc
```

```
):LOCC
-->
:Loc.Location= {rule="LocationRule10"},:Loc.location= {rule=↩
    "LocationRule10"}

Phase:Location4
Input:  Lookup
Options: control = appelt Debug=true

Rule: LocationRule9
Priority:11
(
({Lookup.majorType=="Continents"}|{Lookup.majorType=="↩
    Country"}|{Lookup.majorType=="City"}|{Lookup.majorType=="↩
    Capital"}|{Lookup.majorType=="State"}):Loc
):LOCC
-->
{
gate.AnnotationSet annSet = (gate.AnnotationSet)bindings.get↩
    ("Loc");
gate.FeatureMap features = Factory.newFeatureMap();
try{
int before = annSet.firstNode().getOffset().intValue();
int after = annSet.lastNode().getOffset().intValue();
if(before > 0)
 before--;
String content =doc.getContent().toString().substring(before↩
    ,before+1);
String aftercontent = " ";
    try
    {
        aftercontent =doc.getContent().toString().substring(↩
            after,after+1);
    }
    catch(Exception ex) {}
```

```
    if ( content.equals("ل") || content.equals("ب") || ↩
       content.equals("و") || content.equals(" ") || before ↩
       == 0)
    {
        if( aftercontent.equals(".") || aftercontent.equals(↩
           " ") ||aftercontent.equals("") )
        {
            features.put("rule", "LocationRule9");
            outputAS.add(annSet.firstNode(), annSet.lastNode↩
               (), "location", features);
            outputAS.add(annSet.firstNode(), annSet.lastNode↩
               (), "Location", features);
        }
        else if (aftercontent.equals("ى") || aftercontent.↩
           equals("ي"))
        {
            after++;
            aftercontent =doc.getContent().toString().↩
               substring(after,after+1); //added +1
            if (aftercontent.equals("ة")    )
            {
                features.put("rule", "LocationRule9");
                outputAS.add(annSet.firstNode(), annSet.↩
                   lastNode(), "location", features);
                outputAS.add(annSet.firstNode(), annSet.↩
                   lastNode(), "Location", features);
            }
            else if(aftercontent.equals(" ") || aftercontent↩
               .equals("."))
            {
                features.put("rule", "LocationRule9");
                outputAS.add(annSet.firstNode(), annSet.↩
                   lastNode(), "location", features);
                outputAS.add(annSet.firstNode(), annSet.↩
                   lastNode(), "Location", features);
```

```
            }
        }
    }

}catch(Exception ioe){
      //this should never happen
      throw new GateRuntimeException(ioe);
    }
}


Rule: LocationRule8Post
Priority:15
(
({Lookup.majorType==”Country”}):Loc
{Lookup.majorType==”CountryPost”}
):LOCC
-->
:Loc.Location= {rule=”LocationRule8Post”},:Loc.location= {↩
   rule=”LocationRule8Post”}

Rule: LocationRule8Pre
Priority:15
(
{Lookup.majorType==”CountryPre”}
({Lookup.majorType==”Country”}):Loc
):LOCC
-->
:Loc.Location= {rule=”LocationRule8”},:Loc.location= {rule=”↩
   LocationRule8Pre”}


Phase:Location3
Input:  Lookup Token
Options: control = appelt Debug=true
```

```
Rule: LocationRule7Post
Priority:15
(
({Lookup.majorType==”City”}|{Token}):Loc
{Lookup.majorType==”CityPost”}
):LOCC
-->
:Loc.Location= {rule=”LocationRule7Post”},:Loc.location= {rule=”LocationRule7Post”}

Rule: LocationRule7Pre
Priority:15
(
{Lookup.majorType==”CityPre”}
({Lookup.majorType==”City”}|{Token}):Loc
):LOCC
-->
:Loc.Location= {rule=”LocationRule7”},:Loc.location= {rule=”LocationRule7Pre”}

Rule: LocationRule6
Priority:14
(
{Lookup.majorType==”AdmDiv”}
({Lookup.majorType==”Country”}|{Lookup.majorType==”State”}|{Token})
:Loc
):LOCC
-->
:Loc.Location= {rule=”LocationRule6”},:Loc.location= {rule=”LocationRule6”}

Phase:Location2
Input:  Lookup Token
Options: control = brill Debug=true
```

```
Rule: LocationRule5
Priority:13
(
({Token.string==" بمدينة "}|{Token.string==" مدينة "})
({Lookup.majorType==" City "})
:City
):LOCC
-->
:City.Location= {rule=" LocationRule5 "},:City.location= {rule=" LocationRule5 "}

Rule: LocationRule4
Priority:12
(
{Token.string==" عاصمتها "}
({Token})
:AnyCapital
):LOCC
-->
:AnyCapital.Location= {rule=" LocationRule4 "},:AnyCapital.location= {rule=" LocationRule4 "}


Rule: LocationRule3
Priority:11
(
({Token.string==" عاصمة "}|{Token.string==" العاصمة "}|{Token.string==" لعاصمة "}|{Token.string==" بعاصمة "}|{Token.string==" للعاصمة "}|{Token.string==" بالعاصمة "})
({Lookup.majorType==" Country "})
:Cont
):LOCC
-->
:Cont.Location= {rule=" LocationRule3 "},:Cont.location= {rule=" LocationRule3 "}
```

```
Rule: LocationRule2
Priority:10
(
({Token.string==" عاصمة"}|{Token.string==" العاصمة"}|{Token.string↩
   ==" لعاصمة"}|{Token.string==" بعاصمة"}|{Token.string==" للعاصمة"↩
   }|{Token.string==" بالعاصمة"})
({Lookup.majorType==" Country"})
?
(
{Token}
):Loc
):LOCC
-->
:Loc.Location = {rule=" LocationRule2"},:Loc.location = {rule↩
   =" LocationRule2"}

Phase:Location
Input:  Lookup
Options: control = appelt Debug=true

Rule: LocationRule1
Priority:10
(
({Lookup.majorType==" Direction1"}|{Lookup.majorType=="↩
   Direction2"}|{Lookup.majorType==" Direction3"}|{Lookup.↩
   majorType==" Direction4"}):Dir
(
{Lookup.majorType==" Country"}
|{Lookup.majorType==" City"}|{Lookup.majorType==" State"}|{↩
   Lookup.majorType==" Capital"}):Loc
):LOCC
-->  s
{
```

```
gate.AnnotationSet annSet2 = (gate.AnnotationSet)bindings.↩
   get("Dir");
gate.AnnotationSet annSet = (gate.AnnotationSet)bindings.get↩
   ("Loc");

gate.FeatureMap features = Factory.newFeatureMap();
try{
boolean toAdd=false;
int before = annSet.firstNode().getOffset().intValue();
int after = annSet.lastNode().getOffset().intValue();
int afterDir = annSet2.lastNode().getOffset().intValue();
if(before >0)   before--;
String content =doc.getContent().toString().substring(before↩
   ,before+1);
    if(content.equals("ل"))
    {
        before--;
        content =doc.getContent().toString().substring(↩
           before,before+1);
        if(content.equals("ا"))
        {
            before--;
            content =doc.getContent().toString().substring(↩
               before,before+1);
        }
    }
if(content.equals(" ")  )
{
    if(before == afterDir)
    {
        content = "";
        try
        {
            content =doc.getContent().toString().substring(↩
               after,after+1);
```

```
            if(content.equals("ي") || content.equals("ى") )
                content =doc.getContent().toString().↩
                   substring(after+1,after+2);
            if(content.equals(" ") ||content.equals(".") ||↩
               content.equals("") )
                toAdd=true;
        }
        catch(Exception ex) {}
    }
}
if ( toAdd)
{
        features.put("rule", "LocationRule1");
        outputAS.add(annSet.firstNode(), annSet.lastNode(), ↩
           "location", features);
        outputAS.add(annSet.firstNode(), annSet.lastNode(), ↩
           "Location", features);
}

// create FeatureMap to hold new features
}catch(Exception ioe){
      //this should never happen
      throw new GateRuntimeException(ioe);
    }
}
```

Appendix B

Integration of Gate Application with Java Server Pages - Web Application

B.1 Embedding Gate Application with Java Application

Consider a Gate Application Stored as NERA.gapp. Following Java Code embeds the Gate application as Java Application. We build it as jar file to utilize in Web Application.

Listing B.1: Code to Integrate GATE Application with JAVA Application

```
public class NERARun
{
    private static File gappFile = null;
    private static List annotTypesToWrite = null;
    private static String encoding = null;
    public static void doRun(String FileName) throws ↩
       Exception
    {
        fillOptions();
```

```
System.setProperty("gate.home", "C:\\Program Files\\GATE-5.1"); // Modify Path to point Gate Installation Directory
Gate.init();
CorpusController application = (CorpusController)PersistenceManager.loadObjectFromFile(gappFile);
Corpus corpus = Factory.newCorpus("BatchProcessApp Corpus");
application.setCorpus(corpus);
File docFile = new File(FileName);      // load the document
System.out.print("Processing document " + docFile + "...");
Document doc = Factory.newDocument(docFile.toURL(), encoding);
corpus.add(doc);//put the document in the corpus

try{application.execute();}catch(Exception ex){}    // run the application
corpus.clear();     // remove the document from the corpus again
String docXMLString = null;
if(annotTypesToWrite != null)
{
    Set annotationsToWrite = new HashSet();
    AnnotationSet defaultAnnots = doc.getAnnotations("NE");
    Iterator annotTypesIt = annotTypesToWrite.iterator();
    while(annotTypesIt.hasNext()) {
        AnnotationSet annotsOfThisType =
            defaultAnnots.get((String)annotTypesIt.next());
        if(annotsOfThisType != null) {
            annotationsToWrite.addAll(annotsOfThisType);
```

```
            }
        }
        docXMLString = doc.toXml(annotationsToWrite);
                    // create the XML string using these
           annotations
    }
    // otherwise, just write out the whole document as
       GateXML
    else docXMLString = doc.toXml();
    Factory.deleteResource(doc);// Release the document,
        as it is no longer needed
    String outputFileName = docFile.getName() + ".out.
       xml";     // output the XML to <inputFile>.out.
       xml
    File outputFile = new File(docFile.getParentFile(),
       outputFileName);

    FileOutputStream fos = new FileOutputStream(
       outputFile);        // Write output files using
       the same encoding as the original
    BufferedOutputStream bos = new BufferedOutputStream(
       fos);
    OutputStreamWriter out;
    if(encoding == null)  out = new OutputStreamWriter(
       bos);
    else out = new OutputStreamWriter(bos, encoding);
    out.write(docXMLString);
    out.close();
    System.out.println("done");
    System.out.println("All done");
}
private static void fillOptions() throws Exception
{
    if(annotTypesToWrite == null) annotTypesToWrite =
       new ArrayList();
    annotTypesToWrite.add("Person");
```

```
        annotTypesToWrite.add("Location");
        annotTypesToWrite.add("Organization");
        gappFile = new File("C:\\Users\\6915\\Desktop\\↩
           research\\NERA.gapp"); //Modify Path pointing to ↩
           Save GATE application
        encoding = "UTF-8";
        if(gappFile == null) System.err.println("No .gapp ↩
           file specified");
    }
}
```

B.2 JSP Web Application Code

Add the Jar file produced by code in previous section in"'Web-Inf/lib" folder. Use following code snippet to annotate contents of Arabic text.

Listing B.2: CSS for coloring annotations

```
person
{
    background-color: fuchsia;
    font-weight: bold;
}
location
{
    background-color: lime;
    font-weight: bold;
}
organization
{
     background-color: yellow;
     font-weight: bold;
}
```

Listing B.3: Code Script for JSP Web application

```
<%
 if(s!=null) //s is string parameter holding arabic text ↩
    from any control
  {
     BufferedWriter bw=new BufferedWriter(new ↩
        OutputStreamWriter
                    (new FileOutputStream("C:\\ai project\\↩
                       corpustrial.html"),"UTF-8")) ;
     bw.write(s);
     bw.close();
     NERA.NERARun.doRun("C:\\ai project\\corpustrial.html");

    BufferedReader br=new BufferedReader(new ↩
       InputStreamReader
            (new FileInputStream("C:\\ai project\\↩
               corpustrial.html.out.xml"),"UTF-8" ));
     while( (s=br.readLine()) != null)
     {
                          %>
    <%=s%>
                          <%
     }
 }
%>
```

Appendix C

Machine Learning related Code

C.1 Parse Corpus

Following Java Code Parse the Corpus in XML format and build Vectors, the same is used for parsing Actuall tagged corpus and Corpus Tagged with NERA or GERA rule based systems.

Listing C.1: Code to Parse Corpus and build Vectors

```
private static void GenerateVectorsFromFile(String FileName,↩
   Vector<String> myData,Vector<Integer> myClasses)
{
    try
    {
        DocumentBuilderFactory factory = ↩
           DocumentBuilderFactory.newInstance();
        factory.setValidating(false);
        DocumentBuilder builder = factory.newDocumentBuilder↩
           ();
        Document doc = builder.parse(new File(FileName));
        int cccount=0;
        NodeList Head = doc.getChildNodes(); //Only one top ↩
           node NE
        for(int NumNE=0;NumNE<Head.getLength();NumNE++)
```

```
        {
            NodeList NE = Head.item(NumNE).getChildNodes();
            for(int childs=0;childs<NE.getLength();childs++)
            {
                Node Child = NE.item(childs);
                for (int i=0;i<Child.getChildNodes().↩
                   getLength();i++)
                {
                    String[] Tokens = Child.getChildNodes().↩
                       item(i).getTextContent().split(" ");;
                    for(int tokenindex = 0;tokenindex<Tokens↩
                       .length;tokenindex++)
                    {
                        if( Tokens[tokenindex].trim().length↩
                           () > 1 || (!Tokens[tokenindex].↩
                           equals(" ") && !Tokens[tokenindex↩
                           ].equals("    ") && !Tokens[↩
                           tokenindex].equals("") ) )
                        {
                            cccount++;
                            myData.add(Tokens[tokenindex]);
                            myClasses.add(getValueForClass(↩
                               Child.getChildNodes().item(i)↩
                               .getNodeName()) )    ;
                        }
                        else {
                            if(!stri.contains(Tokens[↩
                               tokenindex]))
                                stri+=Tokens[tokenindex];
                        }
                    }
                }
            }
        }
    }
    catch (Exception e)
```

```
    {
        e.printStackTrace();
    }
}
```

C.2 Generate Training Dataset

Following Java Code generates Training Dataset in CSV format that can be exported to WEKA Machine Learning Software.

Listing C.2: Generate Training Dataset

```
private static void GenerateTrainingStringCSV(Vector<String>↩
    myTrainingData,Vector<Integer> myTrainingClasses,Vector<↩
   Integer> myTrainingClassesCrossValidation, String ↩
   OutputFileName,Vector<Integer> myDataPOSTagsVector,↩
   boolean append)
    {
        BufferedWriter bw;
        try {
            bw = new BufferedWriter(new FileWriter(↩
               OutputFileName,append));
            String TrainingString = "";
            bw.write("NMinusTwo,NMinusOne,LengthGreaterThree↩
               ,N,NPlusOne,NPlusTwo,isPOSNoun,POS,↩
               isLookupPerson,isLookupOrganization,↩
               isLookupLocation" +
                ",leftNeighbourPersonLookup,↩
                   leftNeighbourOrganizationLookup,↩
                   leftNeighbourLocationLookup,↩
                   rightNeighbourPersonLookup,↩
                   rightNeighbourOrganizationLookup,↩
                   rightNeighbourLocationLookup" +
                ",leftNeighbourNearNamedEntity,↩
                   leftNeighbourNearNamedEntityID,↩
```

```
                rightNeighbourNearNamedEntity,↩
                rightNeighbourNearNamedEntityID,↩
                isLeftorRightDot"+
            ",Prefix,Suffix,PrefixTwo,SuffixTwo" +
            ",ACTUAL"+"\r\n");//
        if(myTrainingData.size() > 1)
        {
            int TrainingDataSize = myTrainingData.size();
            String AttributeIndex = ",";
            TrainingString = ""+"" + "0"
            +"" + (AttributeIndex) + "0"
            +"" + (AttributeIndex) + (myTrainingData.get(0).↩
                length()> 3 ? "1":"0")
            +"" + (AttributeIndex) + myTrainingClasses.get↩
                (0);
            if(TrainingDataSize>1)  TrainingString += "" + (↩
                AttributeIndex) + myTrainingClasses.get(1);
            else TrainingString += "" + (AttributeIndex) + "↩
                0";
            if(TrainingDataSize>2) TrainingString += "" + (↩
                AttributeIndex) + myTrainingClasses.get(2);
            else TrainingString += "" + (AttributeIndex) + "↩
                0";
            TrainingString += "" + (AttributeIndex) + ↩
                isPOSTagNoun(myDataPOSTagsVector.get(0));
            TrainingString += "" + (AttributeIndex) + ↩
                myDataPOSTagsVector.get(0);
            TrainingString += "" + (AttributeIndex) + ↩
                getLookupValuePerson(myTrainingData.get(0)); ↩
                //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
                getLookupValueOrganization(myTrainingData.get↩
                (0)); //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
                getLookupValueLocation(myTrainingData.get(0))↩
                ; //added Lookup
```

```
            TrainingString += "" + GenerateMiscCSV(↩
               myTrainingData, 0);
            TrainingString += "" + (AttributeIndex) + ↩
               getClassesForValues(↩
               myTrainingClassesCrossValidation.get(0));
            TrainingString += "\r\n";
            bw.write(TrainingString);
            TrainingString = ""
            +"" + "" + "0"
            +"" + (AttributeIndex) + "" + myTrainingClasses.↩
               get(0)
            +"" + (AttributeIndex) + "" + (myTrainingData.↩
               get(1).length()> 3 ? "1":"0")
            +"" + (AttributeIndex) + "" + myTrainingClasses.↩
               get(1);
            if(TrainingDataSize>2) TrainingString += "" + (↩
               AttributeIndex) + "" + myTrainingClasses.get↩
               (2);
            else TrainingString += "" + (AttributeIndex) + "↩
               " + "0";
            if(TrainingDataSize>3) TrainingString += "" + (↩
               AttributeIndex) + "" + myTrainingClasses.get↩
               (3);
            else TrainingString += "" + (AttributeIndex) + "↩
               " + "0";
            TrainingString += "" + (AttributeIndex) + ↩
               isPOSTagNoun(myDataPOSTagsVector.get(1));
            TrainingString += "" + (AttributeIndex) + "" + ↩
               myDataPOSTagsVector.get(1);
            TrainingString += "" + (AttributeIndex) + ↩
               getLookupValuePerson(myTrainingData.get(1)); ↩
               //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
               getLookupValueOrganization(myTrainingData.get↩
               (1)); //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
```

```
            getLookupValueLocation(myTrainingData.get(1))↩
            ; //added Lookup
        TrainingString += "" + GenerateMiscCSV(↩
            myTrainingData, 1);
        TrainingString += "" + (AttributeIndex) + "" + ↩
            getClassesForValues(↩
            myTrainingClassesCrossValidation.get(1));
        TrainingString += "\r\n";
        bw.write(TrainingString);
        TrainingString = ""
        + "" + myTrainingClasses.get(0)
        + "" + (AttributeIndex) + "" + myTrainingClasses↩
            .get(1)
        + "" + (AttributeIndex) + "" + (myTrainingData.↩
            get(2).length()> 3 ? "1":"0")
        +"" + (AttributeIndex) + "" + myTrainingClasses.↩
            get(2);
        if(TrainingDataSize>3)
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(3);
        else TrainingString += "" + (AttributeIndex) + "↩
            " + "0";
        if(TrainingDataSize>4)
                TrainingString += "" + (AttributeIndex) ↩
                    + "" + myTrainingClasses.get(4);
        else TrainingString += "" + (AttributeIndex) + "↩
            " + "0";
        TrainingString += "" + (AttributeIndex) + ↩
            isPOSTagNoun(myDataPOSTagsVector.get(2));
        TrainingString += "" + (AttributeIndex) + "" + ↩
            myDataPOSTagsVector.get(2);
        TrainingString += "" + (AttributeIndex) + ↩
            getLookupValuePerson(myTrainingData.get(2)); ↩
            //added Lookup
        TrainingString += "" + (AttributeIndex) + ↩
            getLookupValueOrganization(myTrainingData.get↩
```

```
            (2)); //added Lookup
        TrainingString += "" + (AttributeIndex) + ↩
            getLookupValueLocation(myTrainingData.get(2))↩
            ; //added Lookup
        TrainingString += "" + GenerateMiscCSV(↩
            myTrainingData, 2);
        TrainingString += "" + (AttributeIndex) + "" + ↩
            getClassesForValues(↩
            myTrainingClassesCrossValidation.get(2));
        TrainingString += "\r\n";
        bw.write(TrainingString);
        int i=3;
        for(;i<TrainingDataSize-3;i++)
        {
            //AttributeIndex=1;
            TrainingString = "";
            TrainingString += "" + myTrainingClasses.get↩
                (i-2);
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(i-1);
            TrainingString += "" + (AttributeIndex) + ""↩
                + (myTrainingData.get(i).length()> 3 ? "↩
                1":"0");
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(i);
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(i+1);
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(i+2);
            TrainingString += " " + (AttributeIndex++) +↩
                ": 0";// + myTrainingClasses.get(i-2)
            TrainingString += "" + (AttributeIndex) + ↩
                isPOSTagNoun(myDataPOSTagsVector.get(i));
            TrainingString += "" + (AttributeIndex) + ""↩
                + myDataPOSTagsVector.get(i);
            TrainingString += "" + (AttributeIndex) + ↩
```

```
                getLookupValuePerson(myTrainingData.get(i↩
                )); //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
                getLookupValueOrganization(myTrainingData↩
                .get(i)); //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
                getLookupValueLocation(myTrainingData.get↩
                (i)); //added Lookup
            TrainingString += "" + GenerateMiscCSV(↩
                myTrainingData, i);
            TrainingString += "" + (AttributeIndex) + ""↩
                + getClassesForValues(↩
                myTrainingClassesCrossValidation.get(i));
            TrainingString += "\r\n";
            bw.write(TrainingString);
        }
        TrainingString = ""
        + "" + myTrainingClasses.get(i-2)
        +"" + (AttributeIndex) + "" + myTrainingClasses.↩
            get(i-1)
        +"" + (AttributeIndex) + "" + (myTrainingData.↩
            get(i).length()> 3 ? "1":"0")
        +"" + (AttributeIndex) + "" + myTrainingClasses.↩
            get(i);
        if(i+1< TrainingDataSize)
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(i+1);
        else TrainingString += "" + (AttributeIndex) + "↩
            " + "0";
        if(i+2< TrainingDataSize)
            TrainingString += "" + (AttributeIndex) + ""↩
                + myTrainingClasses.get(i+2);
        else TrainingString += "" + (AttributeIndex) + "↩
            " + "0";
        TrainingString += "" + (AttributeIndex) + ↩
            isPOSTagNoun(myDataPOSTagsVector.get(i));
```

```
TrainingString += "" + (AttributeIndex) + "" + ↩
    myDataPOSTagsVector.get(i);
TrainingString += "" + (AttributeIndex) + ↩
    getLookupValuePerson(myTrainingData.get(i)); ↩
    //added Lookup
TrainingString += "" + (AttributeIndex) + ↩
    getLookupValueOrganization(myTrainingData.get↩
    (i)); //added Lookup
TrainingString += "" + (AttributeIndex) + ↩
    getLookupValueLocation(myTrainingData.get(i))↩
    ; //added Lookup
TrainingString += "" + GenerateMiscCSV(↩
    myTrainingData, i);
TrainingString += "" + (AttributeIndex) + "" + ↩
    getClassesForValues(↩
    myTrainingClassesCrossValidation.get(i));
TrainingString += "\r\n";
bw.write(TrainingString);
i++;
+" " + (AttributeIndex++) + ":" + ↩
    myTrainingClasses.get(i-3)
+ "" + myTrainingClasses.get(i-2)
+ "" + (AttributeIndex) + "" + myTrainingClasses↩
    .get(i-1)
+" " + (AttributeIndex++) + ":" + (↩
    myTrainingData.get(i-1).length()> 3 ? "1":"0"↩
    )
+"" + (AttributeIndex) + "" + (myTrainingData.↩
    get(i).length()> 3 ? "1":"0")
+"" + (AttributeIndex) + "" + myTrainingClasses.↩
    get(i)
+"" + (AttributeIndex) + "" + myTrainingClasses.↩
    get(i+1);
TrainingString += "" + (AttributeIndex) + "" + "↩
    0";
TrainingString += "" + (AttributeIndex) + ↩
```

```
            isPOSTagNoun(myDataPOSTagsVector.get(i));
        TrainingString += "" + (AttributeIndex) + "" + ↩
            myDataPOSTagsVector.get(i);
        TrainingString += "" + (AttributeIndex) + ↩
            getLookupValuePerson(myTrainingData.get(i)); ↩
            //added Lookup
        TrainingString += "" + (AttributeIndex) + ↩
            getLookupValueOrganization(myTrainingData.get↩
            (i)); //added Lookup
        TrainingString += "" + (AttributeIndex) + ↩
            getLookupValueLocation(myTrainingData.get(i))↩
            ; //added Lookup
        TrainingString += "" + GenerateMiscCSV(↩
            myTrainingData, i);
        TrainingString += "" + (AttributeIndex) + "" + ↩
            getClassesForValues(↩
            myTrainingClassesCrossValidation.get(i));
        TrainingString += "\r\n";
        bw.write(TrainingString);
        i++;
        TrainingString = ""
        +"" + myTrainingClasses.get(i-2)
        +"" + (AttributeIndex) + "" + myTrainingClasses.↩
            get(i-1)
        +"" + (AttributeIndex) + "" + (myTrainingData.↩
            get(i).length()> 3 ? "1":"0")
        +"" + (AttributeIndex) + "" + myTrainingClasses.↩
            get(i);
        TrainingString += "" + (AttributeIndex) + "" + "↩
            0";
        TrainingString += "" + (AttributeIndex) + "" + "↩
            0";
        TrainingString += "" + (AttributeIndex) + ↩
            isPOSTagNoun(myDataPOSTagsVector.get(i));
        TrainingString += "" + (AttributeIndex) + "" + ↩
            myDataPOSTagsVector.get(i);
```

```
            TrainingString += "" + (AttributeIndex) + ↩
               getLookupValuePerson(myTrainingData.get(i)); ↩
               //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
               getLookupValueOrganization(myTrainingData.get↩
               (i)); //added Lookup
            TrainingString += "" + (AttributeIndex) + ↩
               getLookupValueLocation(myTrainingData.get(i))↩
               ; //added Lookup
            TrainingString += "" + GenerateMiscCSV(↩
               myTrainingData, i);
            TrainingString += "" + (AttributeIndex) + "" + ↩
               getClassesForValues(↩
               myTrainingClassesCrossValidation.get(i));
            TrainingString += "\r\n";
            bw.write(TrainingString);
        }
        bw.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
}
```

Appendix D

Decision Tree Model Extracts

Decision Tree extracts for J48 classifier built on ANERcorp Corpus Data using all features for Machine Learning for top node N with values Location, OTHER and Person are shown in Figures D.1, D.2, and D.3 below respectively.

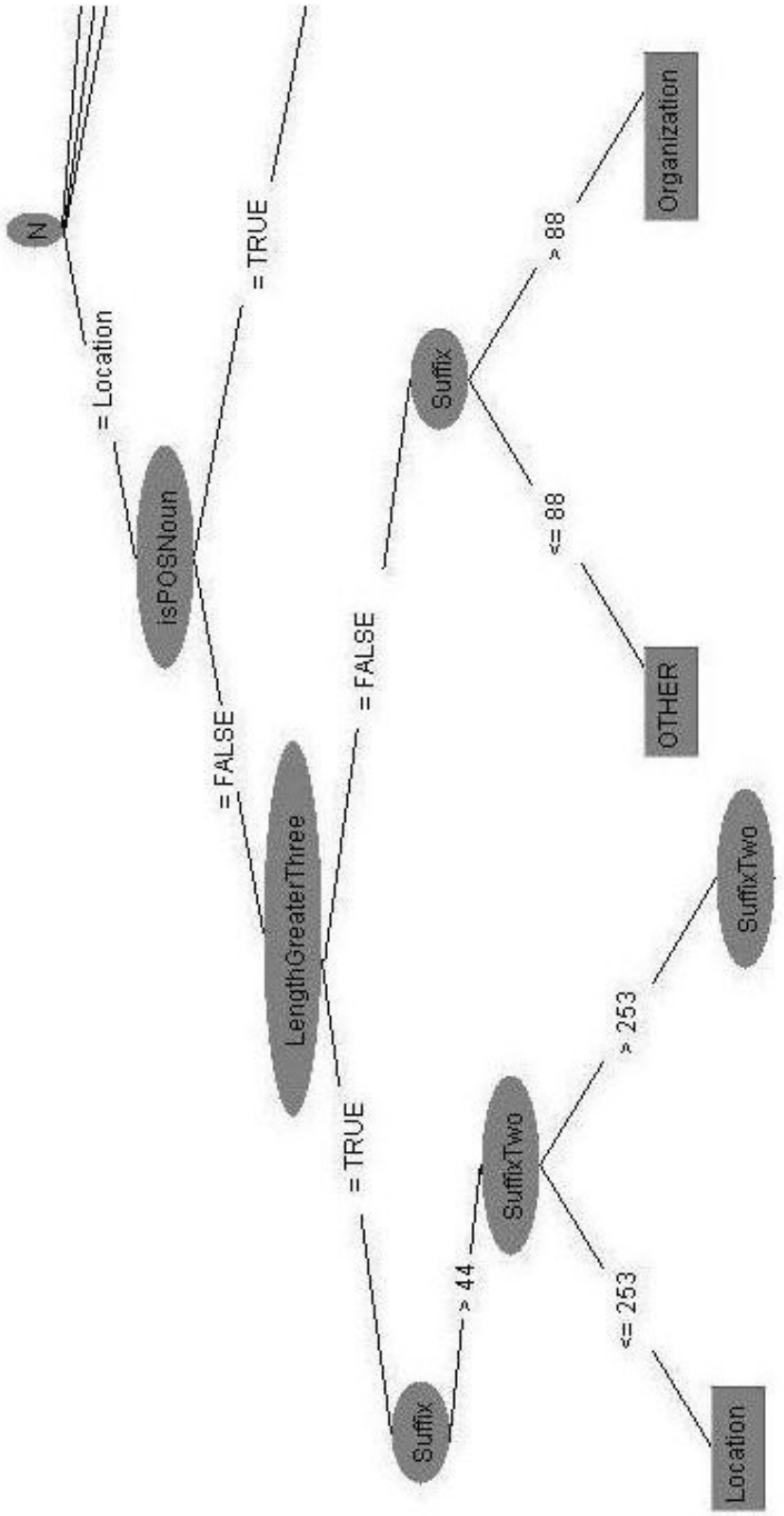

Figure D.1: Subtree for J48 Decision Tree, N=Location - for ANERcorp Corpus using all features

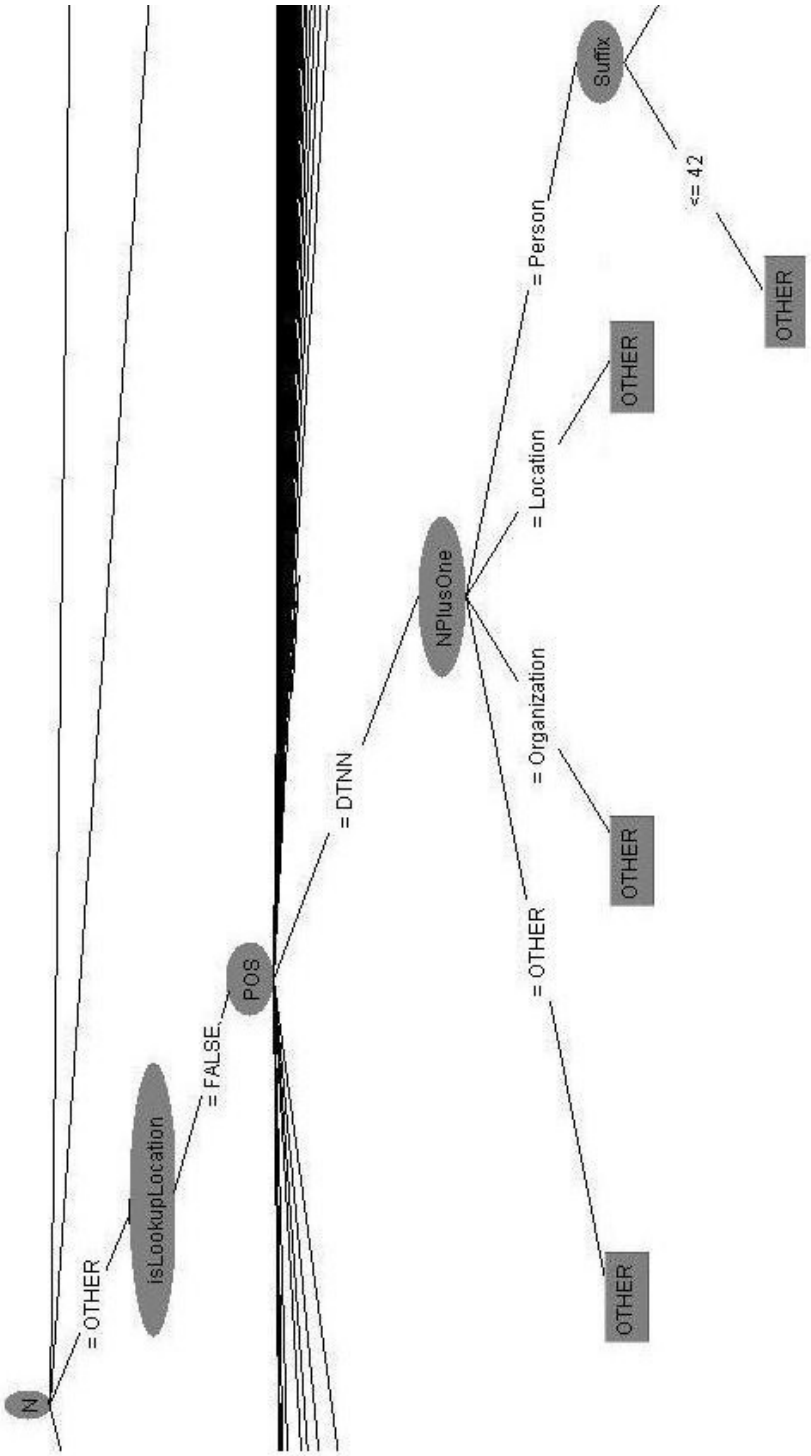

Figure D.2: Subtree for J48 Decision Tree, N=OTHER - for ANERcorp Corpus using all features

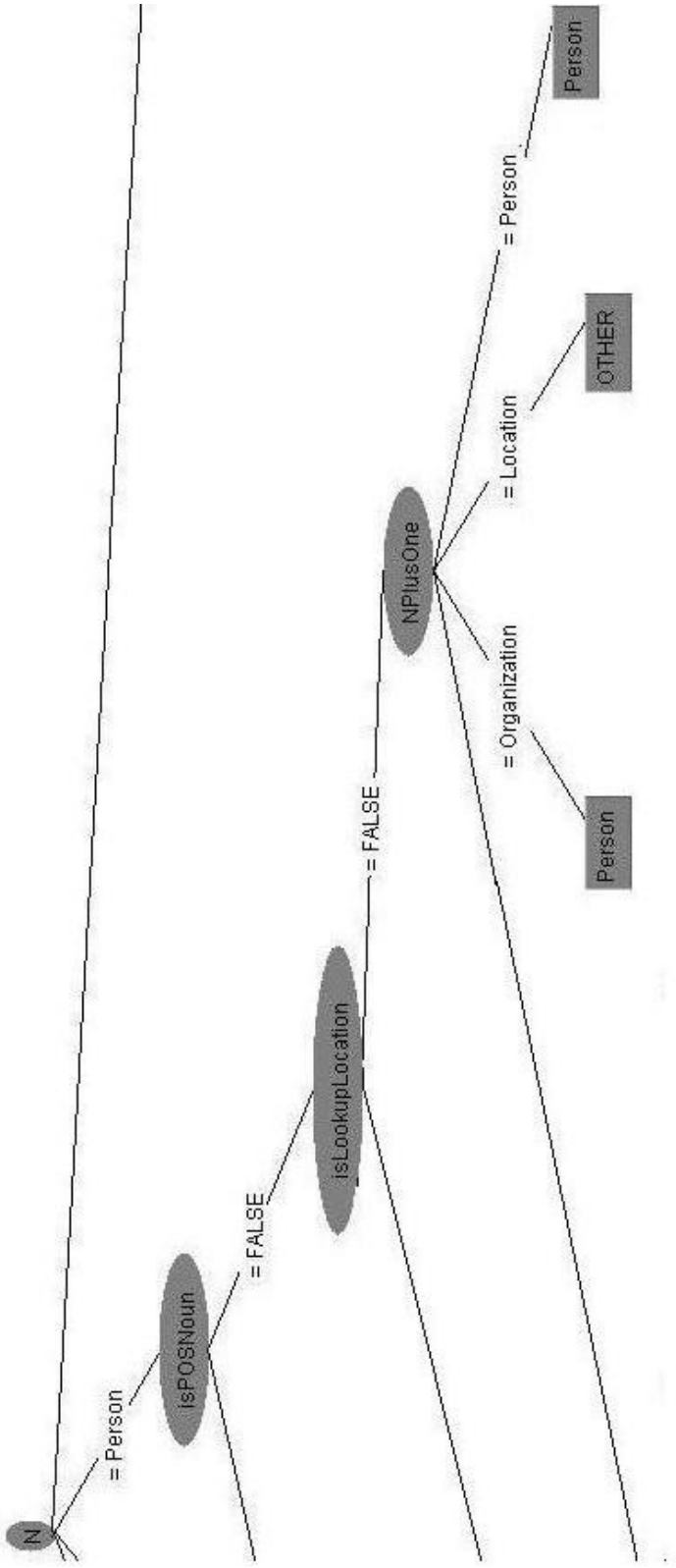

Figure D.3: Subtree for J48 Decision Tree, N=Person - for ANERcorp Corpus using all features

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Muhammad Shoaib)