Integrating Rule-based Approach and Machine learning Approach for Arabic Named Entity Recognition

تكامل منهجية القواعد مع منهجية تعلم الآلة للتعرف على أنماط الأسماء العربية

By
Mai Mohamed Oudah

Dissertation submitted in partial fulfillment of
MSc Informatics (Knowledge and Data Management)

Faculty of Engineering & IT

Dissertation Supervisor
Dr. Khaled Shaalan

May-2012

Abstract

Named Entity Recognition is considered one of the crucial Information Extraction tasks in which many of Natural Language Processing applications rely on as an important preprocessing step. Named Entity Recognition has been successfully applied on different natural languages such as English, French, German, Chinese and Arabic. Natural Language Processing for Arabic has started receiving attention in the past few years as a challenge especially when it comes to information extraction due to the complex nature of Arabic language which rises from the Arabic complicated syntax and rich morphology. However, Named Entity Recognition for Arabic is in its early stages where opportunities for improvement in the performance still available. Most of Arabic NER systems have been developed using mainly two types of approaches including Rule-based approach and Machine Learning based approach.

In this thesis, the problem of Named Entity Recognition for Arabic is tackled through integrating the Machine Learning based approach with the Rule-based approach to form a hybrid approach in attempt to enhance the overall performance of Arabic Named Entity Recognition. The proposed hybrid system is capable of recognizing 11 different types of named entities including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name.

The proposed Arabic named entity recognition system is composed of two main components including a Rule-based component and a Machine Learning based component. The Rule-based component is a reproduction from the acquired linguistic knowledge of the NERA system which has gone through enhancements. The Machine Learning based component utilizes the following techniques: decision trees, support vector machines and logistic regression in order to generate a model for Arabic NER upon an annotated dataset produced by the rule-based component. An annotated dataset is presented to the Machine Learning based component

through a set of features. The feature set is carefully and reasonably selected to optimize the performance of the Machine Learning component as much as possible. Two types of relevant linguistic resources are collected and acquired: gazetteers and corpora (i.e. datasets).

A number of extensive experiments are conducted on three different dimensions including the named entity types, the feature set and the machine learning technique to evaluate the performance of our hybrid Arabic Named Entity Recognition system when applied on different datasets. The experimental results show that the hybrid approach outperforms the Rule-based approach and the Machine Learning based approach separately when it comes to Named Entity Recognition for Arabic. According to the experimental analysis, the best performance of our proposed system is achieved when all the features of different types are considered in the feature set. Decision trees approach has proved its efficiency as a classifier in the proposed hybrid system for Arabic Named Entity Recognition in which the highest overall improvement in the performance is achieved when decision trees approach is used as the classifier. Our hybrid NER system for Arabic outperforms the state-of-the-art of the Arabic Named Entity Recognition in terms of precision, recall and f-measure when applied to ANERcorp dataset with precision of 94.7%, recall of 94.1% and f-measure of 94.4% for Person named entity, precision of 91.7%, recall of 88.6% and f-measure of 90.1% for Location named entities, and precision of 89.4%, recall of 87% and f-measure of 88.2% for Organization named entities.

خلاصة

يعتبر التعرف على أنماط الأسماء أحد أهم العمليات في مجال استخلاص المعلومات حيث تعتمد العديد من تطبيقات معالجة اللغات الطبيعية عليه كخطوة تجهيزية مهمة. لقد تم تطبيق التعرف على أنماط الأسماء في العديد من اللغات الطبيعية مثل الإنجليزية، والفرنسية، والألمانية، والصينية والعربية. وقد بدأت معالجة اللغات الطبيعية للغة العربية تلقى اهتمامًا كبيرًا في السنوات القليلة الماضية باعتبارها تحديًا خصوصًا عندما يتعلق الأمر بمجال استخلاص المعلومات بسبب الطبيعة الغنية والمعقدة للنحو والصرف في اللغة العربية. ومع ذلك فإن التعرف على أنماط الأسماء في اللغة العربية يعد في مراحله الأولى حيث فرص التطوير والتحسين في الأداء لا تزال متاحة. معظم أنظمة التعرف على أنماط الأسماء العربية تم تطويرها باستخدام نوعين من المنهجيات بما في ذلك الأسلوب المبني على القواعد والأسلوب المبني على تعلم الآلة.

في هذه الأطروحة، عملية التعرف على أنماط الأسماء يتم معالجتها من خلال دمج المنهجية المبنية على تعلم الآلة مع منهجية القواعد لتشكيل الأسلوب المُركّب في محاولة لتحسين أداء التعرف على أنماط الأسماء في اللغة العربية. النظام المُركّب المُقترح قادر على التعرف على 11 نوعًا مختلفًا من أنماط الأسماء بما في ذلك أسماء الأشخاص، والأماكن، والمنظمات، والتواريخ، والأوقات، والأسعار (الأموال)، والمقاييس (المقادير القياسية)، والنسب المئوية، وأرقام الهواتف، وردمك (الرقم الدولي المعياري للكتاب)، وأسماء الملفات.

يتألف النظام المُقترح للتعرف على أنماط الأسماء العربية من عنصرين رئيسيين بما في ذلك عنصر يستند على القواعد وعنصر يستند على تعلم الآلة. العنصر الذي يستند على القواعد مبني على معلومات مكتسبة من نظام سابق للتعرف على أنماط الأسماء العربية وقد تم اجراء تحسينات عليها لرفع الأداء. العنصر الذي يستند على تعلم الآلة يستخدم التقنيات التالية: شجرات القرار، مكائن ذات الدعم الموجه، والانحدار اللوجستي وذلك من أجل توليد نموذج للتعرف على أنماط الأسماء العربية بناءً على مجموعة بيانات مُؤشّرة تم انتاجها من قبل العنصر المستند على القواعد. تُعرض مجموعة البيانات المُؤشّرة على العنصر المستند على تعلم الآلة من خلال مجموعة من السمات. وقد تم اختيار مجموعة السمات بعناية وبمنطقية لتحسين أداء العنصر المستند على تعلم الآلة قدر الإمكان. لقد تم جمع نوعين من الموارد اللغوية ذات الصلة: المعاجم (قوائم بالأسماء والكلمات الدلالية) والمجاميع (قواعد البيانات).

وقد تم إجراء العديد من التجارب المكثفة على ثلاثة أبعاد مختلفة بما في ذلك أنواع أنماط الأسماء، ومجموعة السمات، وتقنية تعلم الآلة لتقييم أداء النظام المُركّب عند تطبيقه على مجموعات مختلفة من البيانات. تُظهر النتائج التجريبية تفوق الأسلوب المُركّب على الأسلوب المبني على القواعد وعلى الأسلوب المبني على تعلم الآلة كل على حدة عندما يتعلق الأمر بالتعرف على أنماط الأسماء في اللغة العربية. وفقًا للتحليل التجريبي، يتم تحقيق أفضل أداء لنظامنا المُقترح عند أخذ جميع السمات بمختلف أنواعها بعين الاعتبار في مجموعة السمات. ولقد أثبتت تقنية شجرات القرار فعاليتها عند اتخاذها كمُصنِّف في النظام المُقترح من أجل التعرف على أنماط الأسماء في اللغة العربية حيث يتم من خلالها تحقيق أعلى تحسّن في الأداء العام للنظام المُقترح. يتفوق نظامنا المُركّب على أفضل الأنظمة المنشورة في الدوريات العلمية في مجال التعرف على أنماط الأسماء العربية من حيث الدقة، الشمولية، والمعدل التوافقي بين الدقة والشمولية وذلك عند تطبيق نظامنا على مصادر لغوية البيانات "ANERcorp" بنتيجة دقة قدرها 94.7% وشمولية قدرها 94.1% ومعدل توافقي قدره 94.4% في حالة أسماء الأشخاص، ونتيجة دقة قدرها 91.7% وشمولية قدرها 88.6% ومعدل توافقي قدره 90.1% في حالة أسماء الأماكن، ونتيجة دقة قدرها 89.4% وشمولية قدرها 87% ومعدل توافقي قدره 88.2% في حالة أسماء المنظمات.

Acknowledgements

I am thankful to Allah, for providing me with the well to carry on and never give up. I would like to thank my family for providing support and encouragement throughout my study. I also extend my gratitude to my supervisor, Dr. Khaled Shalaan, for his guidance and support, and for being their when I needed the help.

Declarations

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.

(Mai Mohamed Oudah)

Contents

List of Figures vii

List of Tables ix

1 Introduction 1
1.1 Overview about Named Entity Recognition 1
1.2 Aims and Objectives 2
1.3 Research Questions 3
1.4 Structure of The Thesis 3

2 Named Entity Recognition – Literature Review 5
2.1 Named Entity Recognition and NLP Applications 5
2.1.1 Information Retrieval 6
2.1.2 Machine Translation 6
2.1.3 Question Answering 6
2.1.4 Text Clustering 6
2.2 Arabic Language Characteristics 7
2.2.1 No Capitalization 7
2.2.2 The Agglutinative Nature 7
2.2.3 No Short Vowels 7
2.2.4 Spelling Variants 8
2.2.5 The Arabic Resources 8
2.3 Standard NER Tag Sets 8
2.3.1 The 6th Message Understanding Conference (MUC-6) 8
2.3.2 The Conference on Natural Language Learning (CoNLL) 9
2.3.3 The Automatic Content Extraction program (ACE) 10
2.4 Linguistic Resources 11
2.4.1 Corpora 11
2.4.2 Gazetteers 12
2.5 Named Entity Recognition Approaches 12
2.5.1 Rule-Based NER 13
2.5.2 Machine Learning Based NER 13
2.5.2.1 ML Methods 14

2.5.2.1.1 Maximum Entropy (ME) 14
2.5.2.1.2 Hidden Markov Models (HMM) 14
2.5.2.1.3 Support Vector Machine (SVM) 15
2.5.2.1.4 Conditional Random Fields (CRF) 15
2.5.2.1.5 Decision Trees (DT) 16
2.5.2.2 Feature Set 16
2.5.2.2.1 Word level features 17
2.5.2.2.2 Dictionary-based features 17
2.5.2.2.3 Part-of-speech (POS) tag 17
2.5.2.2.4 Morphological features 17
2.5.2.2.5 Contextual features 17
2.5.3 Hybrid NER 19
2.6 Tools 20
2.6.1 NLP Developmental Environment tools 20
2.6.1.1 GATE 20
2.6.1.2 NooJ 21
2.6.1.3 LingPipe 22
2.6.2 ML tools 22
2.6.2.1 YASMET 22
2.6.2.2 CRF++ 23
2.6.2.3 YamCha 23
2.6.3 Arabic Processing Tools 23
2.6.3.1 BAMA 24
2.6.3.2 MADA 24
2.6.3.3 AMIRA 25
2.6.3.4 Research and development international toolkit (RDI) 26
2.7 Related work 26
2.7.1 Rule-Based NER Systems 27
2.7.2 ML-Based NER Systems 29
2.7.3 Hybrid NER Systems 32
2.8 Conclusion 35

3 Data Collection **36**
3.1 Training and Testing Corpora 36
3.1.1 ACE Corpora 37
3.1.2 ATB Part1 v. 2.0 Dataset 40
3.1.3 ANERcorp Dataset 41
3.1.4 Our Own Corpus 41
3.2 The System's Gazetteers 43
3.2.1 Gazetteers for Person, Location and Organization Extractors 43
3.2.2 Gazetteers for Date, Time, Price, Measurement & Percent Extractors 47
3.2.3 Gazetteers for Phone Number, File Name and ISBN Extractors 53

3.3 Data Verification and Correction 55
3.4 Data Preparation 56
3.5 Conclusion 60

4 The Architecture of the System - The Rule-Based Component 61
4.1 The Proposed Hybrid System 61
4.2 The Implementation of the NE Extractors 63
4.2.1 Processing resources used in GATE 63
4.2.2 Rule-based Named Entity Extractors 64
4.2.2.1 Person NE Extractor 64
4.2.2.2 Location NE Extractor 65
4.2.2.3 Organization NE Extractor 66
4.2.2.4 Date NE Extractor 67
4.2.2.5 Time NE Extractor 70
4.2.2.6 Price NE Extractor 71
4.2.2.7 Measurement NE Extractor 73
4.2.2.8 Percent NE Extractor 74
4.2.2.9 Phone Number NE Extractor 76
4.2.2.10 File Name NE Extractor 77
4.2.2.11 ISBN NE Extractor 79
4.3 Integration of Different NE Extractors 80
4.4 Conclusion 82

5 The Architecture of the System - The ML-Based Component 83
5.1 Machine Learning Approaches Tool – WEKA 83
5.2 The Selected Machine Learning Approaches 83
5.3 The Architecture of the ML-based Component 84
5.4 Feature Set and Feature Extraction 85
5.4.1 General Feature Set 87
5.4.2 Feature set of the 1st group 88
5.4.3 Feature set of the 2nd group 89
5.4.4 The feature set of the 3rd group 91
5.5 Conclusion 93

6 Experimental Analysis 94
6.1 Confusion Matrix 94
6.2 Experimental Setup 95
6.3 Experiments and Results 96
6.4 The Answers to Research Questions 109

7 Enhancing the Grammatical Rules **110**
7.1 Methodology 110
7.2 New Grammatical Rules 111

8 Conclusion and Future Work **113**
8.1 Conclusion 113
8.2 Future Work 117

References **118**

List of Figures

2.1 Sample of ACE 2005 Entity Information 10

3.1 Sample of an ACE 2003 Data File 38
3.2 Sample of an ACE 2003 Entity Information file (Annotations file) 39
3.3 Sample of an ATB data file 40
3.4 Sample of ANERcorp Dataset 41
3.5 Sample of our own corpus 42
3.6 Sample of ACE 2003 NW transformed dataset 58
3.7 Sample of ANERcorp transformed dataset 59

4.1 The Architecture of the Hybrid NER System 62
4.2 Grouping the processing resources under one application in GATE to form the rule-based system 80
4.3 The annotations as appear in GATE interface when the system is applied on ACE 2003 BN dataset – Temporal and Numerical expressions are selected to appear on the GATE interface 81

5.1 The Architecture of the Training Phase 84
5.2 The Architecture of the Prediction Phase 85
5.3 Feature Selection and Extraction Phase 92

6.1 Confusion Matrix 94

List of Tables

2.1 Examples for Arabic Corpora 11
2.2 Examples for free gazetteers 12
2.3 Features of different types used in Arabic ML-based NER systems 18
2.4 Coverage of feature types in Arabic ML-based NER systems 19
2.5 MADA Morphological Features and their Description 25

3.1 The internet resources used to build our own corpus 42
3.2 The gazetteers prepared for the Person names extractor 43
3.3 Examples of entries for each Person names gazetteer 44
3.4 The gazetteers prepared for the Location names extractor 44
3.5 Examples of entries for each Location names gazetteer 45
3.6 The gazetteers prepared for the Organization names extractor 46
3.7 Examples of entries for each Organization names gazetteer 46
3.8 The gazetteers prepared for the Date extractor 47
3.9 Examples of entries for each Date gazetteer 48
3.10 The gazetteers prepared for the Time extractor 49
3.11 Examples of entries for each Time gazetteer 50
3.12 The gazetteers prepared for the Price extractor 50
3.13 Examples of entries for each Price gazetteer 51
3.14 The gazetteers prepared for the Measurement extractor 52
3.15 Examples of entries for each Measurement gazetteer 52
3.16 The gazetteers prepared for the Percent extractor 53
3.17 Examples of entries for each Percent gazetteer 53
3.18 The gazetteers prepared for the Phone Number extractor 54
3.19 Examples of entries for each Phone number gazetteer 54
3.20 The gazetteers prepared for the File name extractor 55
3.21 Examples of errors in the datasets 56
3.22 The distribution of the tag set over the ACE corpora 58
3.23 Number of different NEs annotated for NER in each dataset 60

4.1 The Number of Gazetteers and Rules in each NE Extractor 81

6.1 The results of the Hybrid system evaluation when applied on ACE 2003 NW dataset to recognize the 1st group NEs 97

6.2 The results of the Hybrid system evaluation when applied on ANERcorp dataset to recognize the 1st group NEs 98
6.3 The results of ANERsys 1.0, ANERsys 2.0, CRF-based system and Abdallah, Shaalan and Shoaib (2012)'s system compared to our hybrid system's highest performance when applied to ANERcorp dataset 99
6.4 The results of the Hybrid system evaluation when applied on ACE 2003 BN dataset to recognize the 1st group NEs 99
6.5 The results of the Hybrid system evaluation when applied on ACE 2004 NW dataset to recognize the 1st group NEs 100
6.6 The results of the Hybrid system evaluation when applied on ACE 2003 NW dataset to recognize the 2nd group NEs 101
6.7 The results of the Hybrid system evaluation when applied on ACE 2003 BN dataset to recognize the 2nd group NEs 102
6.8 The results of the Hybrid system evaluation when applied on ACE 2004 NW dataset to recognize the 2nd group NEs 103
6.9 The results of the Hybrid system evaluation when applied on ACE 2004 BN dataset to recognize the 2nd group NEs 104
6.10 The results of the Hybrid system evaluation when applied on ACE 2005 NW dataset to recognize the 2nd group NEs 105
6.11 The results of the Hybrid system evaluation when applied on ACE 2005 BN dataset to recognize the 2nd group NEs 106
6.12 The results of the Hybrid system evaluation when applied on ATB part1 v 2.0 dataset to recognize the 2nd group NEs 107
6.13 The results of the Hybrid system evaluation when applied on our own corpus to recognize the 3rd group NEs 108

7.1 Sample of NEs correctly classified by the hybrid system but misclassified by the rule-based component separately when applied on ACE 2004 NW dataset 110

8.1 The Number of Gazetteers and Rules in each NE Type 114
8.2 Number of different NEs annotated for NER in each dataset 115
8.3 The results of ANERsys 1.0, ANERsys 2.0, CRF-based system and Abdallah, Shaalan and Shoaib (2012)'s hybrid system compared to our hybrid system's highest performance when applied to ANERcorp dataset 116

Chapter 1

Introduction

This chapter gives an overview about Named Entity Recognition and also describes the motivations, goals and objectives of this thesis. The research questions are highlighted along with the structure of this thesis.

1.1. Overview about Named Entity Recognition

Named Entity Recognition (NER) is one of the Natural Language Processing tasks which can be considered an Information Extraction subtask. NER is the task of detecting and classifying named entities (i.e. proper names) within unstructured and structured texts into predefined classes (e.g. person, location and organization) (Nadeau and Sekine, 2007; Shaalan and Raza, 2008). Many of Natural Language Processing (NLP) applications such as machine translation, information retrieval and question answering rely on NER as an important preprocess step. In the literature, three types of approaches are used to develop NER systems including handcrafted rule-based approach, machine learning (ML) based approach and hybrid approach. The rule-based approach relies on handcrafted grammatical rules, while ML-based approach takes advantage of different ML algorithms that utilize sets of features extracted from annotated datasets (i.e. annotated with named entities) for building NER systems. Hybrid approach combines the rule-based approach and the ML-based approach together in order to improve the overall performance of a NER system. NER has been applied on different natural languages such as English, French, German, Chinese and Arabic.

1.2. Aims and Objectives

Arabic language is the official language in the Arabian world where more than 300 million people have Arabic as their native language (Shaalan, 2010). Arabic is one of the Semitic languages and it is the language of the Holy Quran. Thus, every Muslim around the world uses Arabic in daily in their praying. Arabic language, as well as other Semitic Languages, is one of the richest natural languages in the world in terms of morphology and inflection.

NLP for Arabic has started receiving attention in the past few years as a challenge especially when it comes to information extraction due to the complex nature of Arabic language which rises from the Arabic complicated and rich morphology. However, NER for Arabic is in its early stages where opportunities for improvement in the performance still available. A number of Arabic NER systems have been developed using mainly two types of approaches including the rule-based approach, notably NERA system (Shaalan and Raza, 2008), and the ML-based approach, notably ANERsys 2.0 (Benajiba and Rosso, 2007). Rule-based NER systems rely on handcrafted grammatical rules written by language specialists. Therefore, any maintenance or update applied to rule-based systems is labor and time consuming especially if the linguists are not available. On the other hand, ML-based NER systems utilize ML techniques that require large tagged datasets for training and testing purposes. ML-based NER systems are adoptable and updatable with minimal time and effort as long as large datasets are available. The lack of linguistic resources creates a critical obstacle when it comes to Arabic NLP in general and Arabic NER in particular.

In this thesis, the problem of NER for Arabic is tackled through integrating the ML-based approach with the rule-based approach to form a hybrid approach in attempt to enhance the overall performance. To the best of our knowledge, only one recent Arabic NER system (Abdallah, Shaalan and Shoaib, 2012) has adopted the hybrid approach in order to recognize three types of named entities in Arabic texts including Person, Location and Organization. Abdallah, Shaalan and Shoaib (2012) have mentioned the use of only one ML technique (i.e. Decision Trees) within their system. Our research aims to develop a hybrid NER system for Arabic that has the ability to extract 11 different types of named entities including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name. The proposed system is composed of two main

components including a rule-based component and an ML-based component. The rule-based component is a reproduction of a previous rule-based NER system (Shaalan and Raza, 2008) with modifications and additions in order to enhance the component's performance and accuracy. This component is reproduced from the previously acquired Arabic linguistic rules. The ML-based component utilizes ML techniques that were used successfully in similar NER for other languages to generate a model for Arabic NER upon an annotated dataset produced by the rule-based component. An annotated dataset is presented to the ML-based component through a set of features. The feature set is carefully and reasonably selected to optimize the performance of the ML component as much as possible. Two types of linguistic resources are collected and acquired as needed including gazetteers and corpora (i.e. datasets). The data is gone through verification and preparation phases before applying NER. Various experiments are conducted to evaluate the proposed system on different dimensions. The output of the proposed hybrid NER system is exploited to suggest new grammatical rules that may improve the performance of the rule-based component.

1.3. Research Questions

This thesis is trying to answer the following research questions:

- *Which approach of the rule-based, ML-based and hybrid approaches gives the best performance in recognizing named entities in Arabic scripts?*
- *What is the suitable feature set for Arabic NER which leads to the best performance?*
- *Which ML approach may work effectively with a rule-based NER system to form a hybrid system for Arabic NER that improves the overall performance?*

1.4. Structure of The Thesis

The remainder of the thesis is organized as follows. Chapter 2 gives an extensive literature review about Named Entity Recognition. Chapter 3 describes the process followed for data collection. Chapter 4 illustrates the architecture of the proposed NER system and then describes in details the rule-based component. Chapter 5 describes the

ML-based component in details along with the mechanism followed for feature selection and extraction. The conducted experiments and the results are illustrated and discussed in Chapter 6. Chapter 7 illustrates a set of new suggested grammatical rules derived from the output of the proposed hybrid system. Finally, Chapter 8 concludes this thesis and illustrates the future work.

Chapter 2

Named Entity Recognition – Literature Review

This chapter describes the relationship between Named Entity Recognition and Natural Language Processing applications, Arabic language characteristics, the standard Named Entity Recognition Tag Sets used for Arabic and other languages, the linguistic resources, Tools utilized for NER development, the Feature set and Named Entity Recognition different approaches in details with related work for Arabic and other languages as well.

2.1. Named Entity Recognition and NLP Applications

Named Entity Recognition (NER) is considered one of the crucial Information Extraction tasks in which many of Natural Language Processing (NLP) applications rely on as an important preprocess step. NER is the task of extracting named entities (i.e. proper names) from structured and unstructured texts and then being classified into predefined classes (e.g. person, location and organization) (Nadeau and Sekine, 2007; Shaalan and Raza, 2008). In the 1990s, at the Message Understanding Conferences in particular, the task of NER was firstly introduced and given attention by the community of research. Three main NER subtasks were defined at the sixth Message Understanding Conference: ENAMEX which includes Person, Location and Organization entities, TIMEX which refers to temporal expressions, and NUMEX which refers to numerical expressions. Customized NER system may require more sub-divisions (i.e. sub-types) in one or more of the NER subtasks in order to fulfill the system goals and objectives, for example, Location names may have sub-types as City, Country, River, Road, etc.

The performance of NER systems utilized by different NLP systems has a significant impact on the overall performance of these NLP systems which makes the quality of NER systems highly required. The role of NER in NLP applications differs from an application to another. Example of the NLP applications which find the functionalities of NER useful for their purposes are Information Retrieval (IR), Machine Translation (MT), Question Answering (QA) and Text Clustering (TC) (Cowie and Wilks, 1996).

2.1.1. Information Retrieval

IR can be defined as the task of identifying and retrieving relevant documents out of a database of documents according to an input query (Benajiba, Diab and Rosso, 2009a). IR can benefit from NER in two phases: firstly, recognizing the NEs within the query; secondly, recognizing the NEs within the documents in the database and extracting the relevant documents taking into account their classified NEs and how they are related to the query. For example, if the query contains the word "مايكروسوفت" maAykruwsuwft "Microsoft" which represents an Organization named entity, then the documents about Microsoft Corporation will be considered relevant and be retrieved.

2.1.2. Machine Translation

MT is the task of translating a text into another natural language. NEs need special approaches to be translated correctly, hence having a NER system as a part of the MT system is important in order to enhance the performance of the overall system (Babych and Hartley, 2003). In the case of Arabic language, person names can be found as regular words in the language and no orthographic characteristics might distinguish between the two forms, for example, the word "وفاء" wafaA' can be read as noun which means trustfulness and loyalty, and it can also be a person name.

2.1.3. Question Answering

QA can be considered as one of the IR applications but with more sophisticated results. QA systems take questions from the user as inputs and give in return concise and precise answers. NER task can be utilized in the phase of analyzing the question in order to recognize the NEs within the question because that will help later in identifying the relevant documents in the database and then extracting the right answer (Hamadene, Shaheen and Badawy, 2011; Molla, Zaanen and Smith, 2006). For instance, the named entity "الشرق الأوسط" Alšarq AlÂwsaT "Middle East" may be classified as an Organization (i.e. Newspaper) or as a Location according to the context. Hence, the proper classification for "الشرق الأوسط" named entity will help deciding which group of documents should be targeted and searched to find the correct answer.

2.1.4. Text Clustering

TC may exploit NER in ranking the resulted clusters based on the ratio of entities each cluster contains (Benajiba, Diab and Rosso, 2009a). This will enhance the process of analyzing the nature of each cluster and also improve the clustering approach in terms of selected features. For example, Time expressions along with Location named entities can

be utilized as factors that will give an indication of when and where the events mentioned in a cluster of documents have happened which opens the door for new features to be produced and taken in consideration.

The NER task is usually used as a preprocessing step in NLP applications to enhance the overall performance of these systems. Beside the NLP applications previously mentioned, NER may also be utilized by other NLP applications such as Search results clustering, speech recognition, and text-to-speech as well.

2.2. Arabic Language Characteristics

Arabic language is one of the richest natural languages in the world in terms of morphology and inflection. Applying NLP tasks in general and NER task in particular is very challenging when it comes to Arabic language because of its characteristics. The main characteristics of Arabic language that act as challenges for NER task are as follows:

2.2.1. No Capitalization

Capitalization is not a feature of Arabic script unlike several natural languages such as English where a NE usually begins with a capital letter. Therefore, the usage of this orthographic feature (i.e. capitalization) is not an option in Arabic NER. However, the English translation of Arabic words may be exploited in this aspect (Farber et al., 2008).

2.2.2. The Agglutinative Nature

Arabic language is inflectional in a high degree due to its agglutinative nature in which a word may consist of prefix(es), lemma and suffix(es) in different combination and that results in a very complicated morphology (AbdelRahman et al., 2010). For example, the word وبحسناتهم wabiHasanaAtihim *'and by their virtues[fem.]'* according to the Habash-Soudi-Buckwalter transliteration scheme (Habash, Soudi and Buckwalter, 2007), consists of wa *'and'* as a conjuction, bi *'by'* as a preposition (i.e. prefixes), *HsnAt* 'virtues[fem.]' as the stem, and him *'their'* as a possessive pronoun (i.e. suffix) (Benajiba, Diab and Rosso, 2009a).

2.2.3. No Short Vowels

One of the features of Arabic script is the use of diacritics. Most of the Arabic texts, which are written in Modern Standard Arabic, do not include the diacritics, i.e. unvocalized, and a word form in Arabic may refer to two or more different words or meanings according to the context they appear in and their diacritics as well. For instance, رقم may refer to the

word *'number'* if its transliteration is raqam, or the meaning of *'give it a number'* if its transliteration is raq~am. This is considered as a source of ambiguity in Arabic NER.

2.2.4. Spelling Variants

In Arabic script, the word may be spelled differently and still refers to the same word with the same meaning. For example, the word جرام jraAm *'Gram'* can be written as غرام graAm *'Gram'*. The typographic variants should be taken into consideration by the NER system in order to return accurate results.

2.2.5. The Arabic Resources

Most of the available Arabic resources are not free for research purposes and may not be suitable for Arabic NER task due to the absence of NEs annotations on the datasets or the size of the datasets which may not be sufficient. The Arabic gazetteers (i.e. predefined lists of Arabic NEs) are rare as well. Therefore, researchers tend to build their own Arabic resources in order to train and evaluate Arabic NER systems. Standard resources for Arabic NER are rare and that makes comparing the results of different Arabic NER systems not possible unless the systems are evaluated using the same dataset.

2.3. Standard NER Tag Sets

There are three main standard tag sets utilized to annotate data sets of different languages including Arabic in the field of NER:

2.3.1. The 6th Message Understanding Conference (MUC-6)

This conference was the initiation of NER task acknowledgement in the research field. According to MUC-6, named entities are classified into three main subtasks:

- **ENAMEX** (i.e. person, location and organization)
  Example for English:
  <ENAMEX TYPE="ORGANIZATION">Microsoft</ENAMEX>
  Example for Arabic: <ENAMEX TYPE="PERSON">محمد</ENAMEX>
- **NUMEX** (i.e. numerical expressions such as money and percentage)
  Example for English: <NUMEX TYPE="MONEY">$100</NUMEX>
  Example for Arabic: <NUMEX TYPE="MONEY">500 درهم</NUMEX>
- **TIMEX** (i.e. Temporal expressions such as date and time)
  Example for English: <TIMEX TYPE="DATE">18 July 1987</TIMEX>
  Example for Arabic: <TIMEX TYPE="TIME">الثامنة صباحًا</TIMEX>

2.3.2. The Conference on Natural Language Learning (CoNLL)

As a result of CoNLL2002 and CoNLL2003, four categories of NEs are defined including person (PERS), location (LOC), organization (ORG), and miscellaneous (MISC). Miscellaneous refers to other NEs that do not belong to person, location or organization classes. CoNLL follows the IOB format in order to tag named entities in a dataset. The CoNLL annotations are as follows:

B-PERS denotes the beginning of a Person named entity

I-PERS denotes a word inside of a Person named entity

B-LOC denotes the beginning of a Location named entity

I-LOC denotes a word inside of a Location named entity

B-ORG denotes the beginning of an Organization named entity

I-ORG denotes a word inside of an Organization named entity

B-MISC denotes the beginning of a Miscellaneous named entity

I-MISC denotes a word inside of a Miscellaneous named entity

O denotes that the word does not belong to any of the previous classes

For example, the sentence “The author is John Smith” is annotated as below:

O The
O author
O is
B-PERS John
I-PERS Smith

2.3.3. The Automatic Content Extraction program (ACE)

Three categories of named entities have been defined by ACE2003 including person, facility, organization and GPE (i.e. geographical and political entities). Later in ACE 2004 & 2005, another two categories have been added to this tag set: vehicles and weapons. Temporal expressions, which follow TIMEX2 specifications, and Numerical expressions including money, phone number and percentage were covered by ACE 2005 Multilingual Training Corpus. An ACE dataset comes with several files of different types in Standard

Generalized Markup Language; each data file has a matching XML file which represents the entity information (i.e. annotations of named entities within a data file). Figure 2.1 illustrates a sample of entity information in ACE 2005 Multilingual Training Corpus from the Arabic dataset.

Figure 2.1: Sample of ACE 2005 Entity Information

However, the definition of each class/tag may differ with some degree from a tag set to another even if the same class/tag is used by both of them. In this research, we follow a tag set including person, location, organization, date, time, price, measurement, percent, phone number, ISBN and file name annotations. The following are the 11 tags which a named entity will be enclosed with one of them:

<Person>*Entity*</Person>
<Location> *Entity*</Location>
<Organization> *Entity*</Organization>
<Date>*Entity* </Date>
<Time>*Entity* </Time>
<Price>*Entity*</Price>
<Measurement>*Entity*</Measurement>
<Percent>*Entity*</Percent>
<PhoneNumber>*Entity*</PhoneNumber>
<ISBN>*Entity*</ISBN>
<FileName>*Entity*</FileName>

2.4. Linguistic Resources

Linguistic resources are essential for building and/or evaluating NER systems. NER systems take advantage of two types of linguistic resources including Corpora and Gazetteers. The researches tend to build their own linguistic resources due to the lack of linguistic resources for some languages such as Arabic.

2.4.1. Corpora

A corpus (i.e. the singular of corpora) is a very large set of text that may be annotated to serve various NLP tasks. However, a corpus should be provided with annotations to be exploited in the development and the evaluation of ML-based NER systems, while rule-based NER systems need annotated corpora as the gold-standard reference for the evaluation of the performance. A corpus may be genre independent/specific, domain independent/specific and may belong to one natural language (i.e. monolingual) or more (i.e. multilingual). The process of tagging a corpus can be handled in manual, semi-automated or fully-automated manner.

The following are examples for annotated Arabic corpora:

- **ACE 2003 corpus**: This corpus is of two genres; Broadcast News (BN) and Newswire (NW).
- **ACE 2004 corpus**: This corpus is of three genres; BN, NW and Arabic Tree Bank (ATB).
- **ACE 2005 corpus**: This corpus is of three genres; BN, NW and Weblogs (WL).
- **ANERcorp:** This corpus is of one genre; NW and it follows the CoNLL tag set and IOB scheme.

ACE corpora are available under license agreement from LDC (www.ldc.upenn.edu) which is a respective source for linguistic resources, while ANERcorp[1] is available for free. Table 2.1 shows the size in words and the number of named entities within previously listed corpora (Benajiba, Diab and Rosso, 2008a); the numbers represent Arabic contents.

The Corpus	The Size (words)	No. of Named Entities
ACE 2003	55.29K	5,505
ACE 2004	154.12K	11,520
ACE 2005	104.65K	10,218
ANERcorp	174.76K	12989

Table2.1: Examples for Arabic Corpora

[1] Available to download on http://www1.ccls.columbia.edu/~ybenajiba/downloads.html

2.4.2. Gazetteers

Another linguistic resource is the gazetteer. The gazetteers are predefined lists of named entities. Dictionaries and Whitelists may refer to gazetteers as well (Shaalan and Raza, 2008). The contents of a gazetteer should be consistent and belong to only one type of named entity per gazetteer such as Person, Location or Organization. Both rule-based and ML-based NER systems may exploit gazetteers in their architecture as in Shaalan and Raza (2009) and Benajiba, Rosso and Bened´ı (2007). In rule-based systems, gazetteers may be utilized in the construction and implementation of the grammatical rules, for example:

CityName + (? + (CountryName | StateName) +)?

This rule identifies a city name as a Location named entity if the city name exists in the dictionary of city names, and followed by, possibly in parentheses, country or state names exist in their corresponding dictionaries. The following is a sentence with a location which is detected using the previous grammatical rule:

المقر الرسمي في **برشلونة** (إسبانيا) (The official headquarter in **Barcelona** (Spain))

On the other hand, ML-based NER system may exploit gazetteers as features in the features set. For example, a Boolean feature that indicates whether a word exists in the Person names gazetteers or not.

Table 2.2 illustrates some free Arabic gazetteers available on the internet for research purposes as stated by Habash (2010):

The gazetteer	Named Entity's types	Language
ANERgazet[2]	Person, Location, Organization	Arabic
Foreignword.com	location	Include Arabic
FAOTERM(fao.org)	location	Include Arabic

Table 2.2: Examples for free gazetteers

2.5. Named Entity Recognition Approaches

NER revolves around two main goals including the detection of proper names and then the extraction of these names in the form of different predefined classes. Three types of approaches are used to fulfill those two goals including rule-based approach, machine learning (ML) approach and hybrid approach. Rule-based systems for NER rely on

[2] Available for free on http://www1.ccls.columbia.edu/~ybenajiba/downloads.html

linguistic grammars to detect and classify proper names. On the other hand, ML-based systems utilize different learning algorithms to generate statistical models for NE prediction. The third type of systems, i.e. based on hybrid approach, is a combination between the rule-based approach and the ML approach which aims to improve the performance of the NER system.

2.5.1. Rule-Based NER

Rule-based NER systems depend on hand-crafted linguistic rules in the process of identifying named entities within text. Such systems exploit predefined lists (i.e. gazetteers/dictionaries) of named entities and/or NE indicators in the structure of the rules. The rules are usually implemented in the form of regular expressions or finite state transducers (Mesfar, 2007; Shaalan and Raza, 2008). The availability of tagged datasets is essential for the evaluation phase of rule-based NER system but not for the development of the system. The maintenance of rule-based systems is not a straight forward process since linguists need to be available to provide the system with the proper adjustments especially for the rules and the dictionaries (Petasis et al., 2001). Thus, any adjustment to the rule-based system requires labor and time consuming.

2.5.2. Machine Learning Based NER

This type of NER systems takes the advantage of machine learning (ML) algorithms. Machine learning approaches that have been used for NER are distributed among two categories: supervised learning (SL) techniques and semi-supervised learning techniques (Nadeau and Sekine, 2007). The main difference between the two categories is that the SL techniques require the availability of large annotated datasets to be utilized in the training phase in order to learn the models to detect and classify named entities within the input, while the semi-supervised learning techniques do not require having annotated datasets in prior. The most common ML techniques used for NER are the supervised learning techniques which represent the NER problem as a classification task. Support Vector Machines (SVM), Conditional Random Fields (CRF), Maximum Entropy (ME), Hidden Markov Models (HMM) and Decision Trees (DT) are common SL techniques for classification that have been used for NER (Nadeau and Sekine, 2007).

2.5.2.1. ML Methods

The common machine learning techniques utilized in the field of NER in general and Arabic NER in particular are Support Vector Machines (SVM), Conditional Random Fields (CRF), Maximum Entropy (ME), Hidden Markov Models (HMM) and Decision Trees (DT).

2.5.2.1.1. Maximum Entropy (ME)

ME approach is about computing weights to be assigned to each feature in the feature set that represents each element in the problem space. In the training phase, ME classifier observes the outcomes (i.e. the classes), the history and the features of all elements in the space to produce a ME model. In NER, the history represents the information extracted from other words in the training dataset relative to the encountered word. The outcome of an element is determined via computing the conditional probability of each possible outcome using the following formula (Pietra, Pietra and Lafferty, 1997) (1):

$$p(o|h) = \frac{1}{Z(h)} \prod_i \alpha_i^{f_i(h,o)} \qquad (1)$$

Where α_i is the weight of feature f_i, o is the outcome, h is the history and $Z(h)$ is the normalization function. The outcome with the highest probability is assigned as the predicted outcome (i.e. class) for an element (e.g. word in the case of NER). YASMET is a tool to compute the weights of ME model and it has been used in a number of Arabic NER systems which adopt ME as the statistical method.

2.5.2.1.2. Hidden Markov Models (HMM)

The HMM approach relies on estimating the likelihood of elements to belong to a given class. For a given sequence of words W (i.e. the input text in the case of NER), the tag sequence T is generated with the most likely tags for each element in the input sequence in order to maximize the following equation (Zhou and Su, 2002) (2):

$$\log P(T^n|W^n) = \log P(T^n) + \log \frac{P(T^n,W^n)}{P(T^n) \cdot P(W^n)} \qquad (2)$$

The T which optimizes the likelihood is produced as the output of the HMM classifier. HMM has been utilized in the field of NER in which in the input sequence is the sequence of words in a dataset and the output tag sequence is the sequence of predicted named entity classes for the words that appear in the input sequence.

2.5.2.1.3. Support Vector Machine (SVM)

SVM is about estimating a hyperplane which divides the elements in space into two groups (i.e. classes) e.g. class + and class−. The margin between the hyperplane and the nearest element needs to be maximized. The hyperplane is estimated during the training phase through observing the features of each element in the training space with its actual class. Each element is represented with a vector of features and its actual class. The position of an element with respect to the hyperplane decides its predicted class by the SVM model. For example, if the element is on the '+' side of the hyperplane then the predicted class is '+'. The position is calculated using the next equation (Vapnik, 1995) (3):

$$g(x) = \sum_i^N w_i k(x, sv_i) + b \qquad (3)$$

Where sv_i represents support vectors which are the nearest data elements to the hyperplane, N is the number of support vectors, $k(x, sv_i)$ are the kernels for features mapping, w_i represents the weights of all features of an element and b is a constant which is computed in the training phase. SVM has been used widely in NER systems due its robustness to noise and its ability to deal with large feature sets effectively. YamCha is a toolkit developed for training SVM models and it has been used in a number of Arabic NER systems which adopt SVM as the statistical method.

In this thesis, SVM approach is selected as one of the ML methods to be observed and utilized in our hybrid NER system for Arabic.

2.5.2.1.4. Conditional Random Fields (CRF)

CRF is a ML method which utilizes binary features on order to construct undirected graphical models to segment and annotate a sequence of data points. The output of a CRF model is a sequence of labels (i.e. classes) in which the conditional probabilities of each label are maximized. The CRF model can be computed using the following formula (Lafferty, McCallum and Pereira, 2001) (4):

$$p(y|x) = \frac{1}{Z(x)} * \exp\left(\sum_t \sum_k \lambda_k . f_k(y_{t-1}, y_t, x)\right) \qquad (4)$$

Where y is the sequence of labels, x is the sequence of data points, λ_k are the weights of each feature in the feature set and $Z(x)$ is the normalization function. As it can be noted from the previous equation, CRF may be considered as an extension of ME and HMM. CRF has shown its efficiency in NER systems (see section 6.2.3). CRF++ is a tool used to perform CRF method, and it has been utilized in different NER systems for Arabic and other different languages.

2.5.2.1.5. Decision Trees (DT)

Decision trees approach (Orphanos et al., 1999) is one of the ML methods which rely on undirected graphs to induce rules in form of a tree. The internal nodes in a tree represent the features in the feature set, while the leaf nodes represent the classes, and the branches represent the feature values; hence the classification is achieved through traversing the tree. The construction of a decision tree is done via choosing the proper feature at each internal node, generating children nodes for each value of the features, then split the data points over the children, and repeat the previous steps for each child. The common method used for feature selection for each node is Information Gain (IG) as used in ID3 and C4.5 decision trees algorithms. The following is the Information Gain formula (Eid et al., 2011) (5):

$$IG(Y|X) = H(Y) - H(Y|X) \qquad (5)$$

Where Y is a class, X is a feature, and H is the Entropy. The Entropy is measured using the following equation (Eid et al., 2011) (6):

$$H(Y) = -\sum_{i=1}^{k} P(y_i) log_2(P(y_i)) \qquad (6)$$

The feature with the maximum IG is chosen to be represented by the internal node. Decision trees have been used as a classification method in different NLP systems such as NER systems. In this thesis, decision trees approach is selected as one of the ML methods to be observed and utilized in our hybrid NER system for Arabic.

2.5.2.2. Feature Set

In ML-based NER systems, each word in an input text has a corresponding vector of features values. One of the crucial issues when it comes to ML techniques is the feature set. The feature set needs to be selected carefully and reasonably to optimize the performance of the ML component as possible. The feature set is composed of attributes that can be exploited to extract patterns in order to predict the classes of previously unseen data entries. For NER, the feature set may contain features of different types including word level features, dictionary-based features, part-of-speech (POS) tag, morphological features, and contextual features. Following are descriptions of the main types of features.

2.5.2.2.1. Word level features

These features are related to the orthographic nature of each word individually. Case features that involve capitalization, Punctuations, Special characters, and Digits are considered word level features.

2.5.2.2.2. Dictionary-based features

Most of NER systems include gazetteers/dictionaries/keywords lists which can be utilized in the feature set through checking the presence of each word in the dataset within the system's gazetteers/dictionaries/keywords lists.

2.5.2.2.3. Part-of-speech (POS) tag

The POS tag of each word in the dataset is considered an important feature which assists in identifying named entities that are usually nouns or proper nouns. Examples of POS tags are verb, noun, proper noun and adjective.

2.5.2.2.4. Morphological features

These features represent the morphological nature of a word such as affixes, number, and verb tense. Extracting morphological feature can be performed through procedures embedded within grammatical rules or through utilizing a morphological analyzer to generate the morphological features to be used then by the NER system.

2.5.2.2.5. Contextual features

Contextual features can be derived from the context of a document to extract the relationships between previously identified entities and an encountered word within the input document. Taking into account the features of a window of words centered by a candidate word in the recognition process is also considered as contextual features utilization.

Table 2.3 illustrates features of different types with their description including word level features, dictionary-based features, POS features and contextual features which have been used in Arabic ML-based NER systems. The morphological features are illustrated in table 2.5.

Feature Type	Feature	Description
Word level	Word length	The number of characters in a word
	Nationality	Identifying nationalities from the input words
	Word Gloss	Checking whether the word's English translation begins with a capital letter or not
	Word n-gram	The preceding and succeeding words n-gram and character n-gram probability
	Special Markers	The presence of punctuations, digits and special characters in a word
Dictionary-based	Gazetteers	Checking the presence of a word in a gazetteer
POS Feature	POS tag	POS tag of a word
Contextual features	Nationality	Identifying nationalities from the input words
	Base phrase chunks	Identifying syntactic phrases such as noun phrases (NPs) and verb phrases (VPs) within an Arabic text
	Word n-gram	The preceding and succeeding words n-gram and character n-gram probability

Table 2.3: Features of different types used in Arabic ML-based NER systems

In Arabic NER, different combinations of features have been used to construct feature sets for different Arabic NER systems as follows:

- POS tags, base phrase chunks (BPC) (i.e. contextual feature), gazetteers (i.e. dictionary-based feature) and nationality (i.e. word-level and contextual feature) have formed the feature set for Benajiba, Rosso and Bened´ı (2007) and Benajiba and Rosso (2007; 2008)'s Arabic ML-based NER systems.

- Contextual features, lexical features (i.e. word-level features), Gazetteers (i.e. dictionary-based feature), Morphological features, POS-tags, Base Phase chunks (BPC) (i.e. contextual feature), nationality (i.e. word-level and contextual feature) and corresponding English capitalization (i.e. word-level feature) have formed the feature set for Benajiba, Diab and Rosso (2008a; 2008b)'s Arabic ML-based NER system.

- Leading and trailing character n-gram, word position, word length, word unigram probability, the preceding and succeeding words n-gram and character n-gram probability (i.e. word-level and contextual features) have been used to form the feature set for Abdul-Hamid and Darwish (2010)'s Arabic ML-based NER system.

– Word-level features, POS tag, Base Phase Chunks (BPC) (i.e. contextual feature), gazetteers (i.e. dictionary-based feature) and morphological features have formed the feature set for the Arabic ML-based system of AbdelRahman et al. (2010).

– Word-level features, morphological features, contextual features, Gloss Capitalization (i.e. word-level feature), and POS tag have been used to form the feature set for Farber et al. (2008)'s Arabic ML-based NER system.

Feature Type / Arabic NER System	Word Level	Dictionary-based	POS tag	Morphological	Contextual
Benajiba, Rosso and Bened´ı (2007), Benajiba and Rosso (2007; 2008)	✓	✓	✓	×	✓
Benajiba, Diab and Rosso (2008a; 2008b; 2009a; 2009b)	✓	✓	✓	✓	✓
Farber et al. (2008)	✓	×	✓	✓	✓
Abdul-Hamid and Darwish (2010)	✓	×	×	×	✓
AbdelRahman et al. (2010)	✓	✓	✓	✓	✓

Table 2.4: Coverage of feature types in Arabic ML-based NER systems

Table 2.4 summaries the coverage of feature types previously listed Arabic ML-based NER systems. In this research, we have used different types of features to form the feature sets for our ML-based component in the proposed hybrid NER system for Arabic.

2.5.3. Hybrid NER

The third type of NER approaches is the Hybrid approach. The hybrid approach is the integration between the rule-based approach and the ML-based approach. In this thesis, the hybrid approach is adopted in attempt to combine the handcrafted rules and the ML techniques to effectively tackle the task of NER for Arabic language to extract eleven types of named entities (i.e. Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name) from Arabic scripts. The direction of the

processing flow may be from the rule-based system to the ML-based system or vice versa. The main purpose of using hybrid approaches is optimizing the overall performance.

2.6. Tools

In this section, the tools exploited in the development of NER system are illustrated with some details. The tools can be classified according to their functions into three categories including NLP developmental environment tools, ML tools and Arabic processing tools.

2.6.1. NLP Developmental Environment tools

This subsection highlights the NLP developmental environment tools that have been used in NER. Reusable components, which are built-in modules/resources, are usually provided within the framework tool to be used in the construction of NER system, such as Tokenizer and Part-of-Speech (POS) tagger. The user may utilize the built-in modules or build his/her own modules according to the tool's programming policy and then incorporate the new modules in the framework. The common tools described in this category are the GATE, NooJ and LingPipe.

2.6.1.1. GATE

GATE[3] (i.e. General Architecture for Text Engineering) is a free open source application which enables users to build and evaluate applications for various NLP tasks using the different built-in resources and components in multiple languages and domains (GATE, 2012). When it comes to NER, GATE facilitates the development of rule-based NER systems through providing the user with the capability to implement grammatical rules as finite state transducer using JAPE. The following summarizes the main components of GATE (GATE, 2012; Zaidi, Laskri and Abdelali, 2010):

a) **CREOLE:** stands for Collection of Reusable Objects for Language Engineering, is a collection of three types of reusable resources: Firstly, language resources such as Corpora and Lexicons. Secondly, processing resources such as Tokenizers, Parsers and Orthomatchers. Finally, visual resources that enable the developer to design GUI and manipulate the other types of reusable resources.

b) **ANNIE:** stands for A Nearly New Information Extraction system, the main component in GATE for building a NER system which can be represented as a

[3] Available for free download on http://gate.ac.uk/

pipeline that includes a Tokenizer, NE Transducer, Orthomatcher, Sentence Splitter, Gazetteer and POS Tagger. ANNIE's processing resources can be separately imported and incorporated in any language processing system developed under the GATE architecture.

c) **JAPE:** stands for Java Annotation Pattern Engine, a pattern specification language provided by GATE to enable the implementation of grammatical rules as finite state transducers based on regular expressions.

GATE has an Arabic plugin for the recognition of various types of named entities including Person, Location, Organization, Date, Time and Percent. The Arabic plugin is composed of a Tokenizer, Gazetteers, Orthomatcher and grammars as the components of a simple Arabic rule-based NER application built within GATE (GATE, 2012). Elsebai, Meziane and BelKredim (2009) and Maynard et al. (2002) have used GATE framework in their research studies for NER. In this thesis, GATE is adopted as a framework to implement a previous rule-based NER system for Arabic (Shaalan and Raza, 2008). In order to build an Arabic NER system using GATE, a set of processing resources can be used as the components of the system including Arabic tokenizer, JAPE transducers for grammatical rules and gazetteers.

2.6.1.2. NooJ

Nooj[4] is a free linguistic environment based on .NET platform, which is an object-oriented architecture, to enable the reusability of Nooj processing components (NooJ, 2012). Nooj is a language independent and domain independent framework where documents of different file formats, such as Unicode, XML and MS-WORD, can be text processed. The users are allowed to build, debug, update and share linguistic components along with the built-in components. The linguistic components within NooJ are dictionaries, morphological and syntactic grammars. NooJ is capable of interpreting rules written in finite state form or context-free grammar form which facilitates the development of rule-based NER systems. Nooj provides disambiguation technique based on grammars to resolve duplicate annotations.

Mesfar (2007) has developed and incorporated an Arabic module within NooJ architecture. The Arabic module contains a dictionary of verbs, morphological grammars that deal with verb affixation, syntactic grammars, gazetteers and lists of trigger words for various types of named entities such as Person, Location and Organization. The components of the Arabic module work along with some other built-in components within

[4] Available for free download on http://www.nooj4nlp.net

Nooj such as the tokenizer, morphological analyzer and the NER unit. All these components together construct a rule-based NER system for Arabic. In order to build Arabic NER systems using NooJ, a number of components can be used including the built-in tokenizer and morphological analyzer, grammatical rules and gazetteers.

2.6.1.3. LingPipe

LingPipe[5] is a toolkit for text engineering and processing. For research purposes, LingPipe is available for free with limited production abilities; hence obtaining its license agreement is required in order to have complete production capabilities (Alias-i, 2008). LingPipe is language, domain and genre independent. Applications of different language processing tasks can be constructed within LingPipe such as part-of-speech (POS) taggers, Spelling Correction systems, named-entity recognizers, and Word Sense Disambiguation systems. A HMM-based named-entity recognizer (i.e. ML-based system) is provided within LingPipe environment, and the learned models can be evaluated using k-fold cross validation over annotated datasets.

LingPipe NER system has been applied on ANERcorp in order to generate a statistical NER model for Arabic. The details with the results are presented in the toolkit official website http://alias-i.com/lingpipe/. AbdelRahman et al. (2010) have compared their Arabic NER system with the LingPipe NER system in terms of performance on ANERcorp. In order to build an Arabic NER system using LingPipe, an annotated corpus with NE tags using IOB schema can be used along with the proper gazetteers.

2.6.2. ML tools

A ML tool is a tool that enables the application of a specific machine learning technique, such as Maximum Entropy (ME), Conditional Random Fields (CRF) and Support Vector Machine (SVM), in order to perform some NLP tasks. NER task benefits from ML tools in building ML-based NER systems. This sub-section focuses on ML tools that have been utilized in building Arabic NER systems.

2.6.2.1. YASMET

A free toolkit, which is written in C++, for learning Maximum Entropy models (MEM). YASMET estimates the parameters of the model; computes the weights of the ME model (YASMET, 2002). YASMET is designed to handle large set of features efficiently. YASMET can be found at http://www-i6.informatik.rwth-aachen.de/web/Software/YASMET.html.

[5] LingPipe is available on http://alias-i.com/lingpipe/

Benajiba, Rosso and Bened´ı (2007) and Benajiba, Diab and Rosso (2009a) have used YASMET to apply ME approach in Arabic NER systems.

2.6.2.2. CRF++

A free open source toolkit, which is written in C++, for learning Conditional Random Field (CRF) models in order to segment and annotate sequence of data (CRF++, 2012). CRF++ can be utilized in many NLP tasks e.g. Text Chunking and NER, and it can handle large feature sets. CRF++ is available on http://crfpp.sourceforge.net/. Benajiba and Rosso (2008), Benajiba, Diab and Rosso (2008b; 2009a) and Abdul-Hamid and Darwish (2010) have utilized CRF++ to perform CRF approach for Arabic NER.

2.6.2.3. YamCha

A free open source toolkit, which is written in C++, for learning Support Vector Machines (SVM) models (YamCha, 2005). YamCha is text chunker in the first place that can be utilized for NER, POS tagging and partial chunking. YamCha is able of handling large sets of features. YamCha is available for free on http://chasen.org/~taku/software/yamcha/. Benajiba, Diab and Rosso (2008a; 2008b; 2009a; 2009b) have used YamCha to train and test SVM models for Arabic NER.

YASMET, CRF++ and YamCha are language independent, domain independent and have no built-in linguistic resources embedded within their architectures (CRF++, 2012; YamCha, 2005; YASMET, 2002). The ML tools can be used to build Arabic NER systems that have an ML-based component where a specific ML approach is applied on an annotated dataset in order to analyze the features of each entry and generate a statistical model for NER. The feature set may utilize a set of predefined gazetteers as a component of the NER system.

2.6.3. Arabic Processing Tools

An Arabic processing tool is utilized to perform language analysis tasks on Arabic scripts such as morphological analysis, POS tagging, deriving base phrase chunks (BPC) and so on. In this section, we illustrate some of Arabic analysis tools that have been used in the field of Arabic NER:

2.6.3.1. BAMA

BAMA[6] stands for Buckwalter Arabic Morphological Analyzer, which is an Arabic morphological analyzer available through license agreement obtained from Linguistic Data Consortium (LDC). BAMA has three components (Habash , 2010):

a) **Lexicon:** For every data entry, the prefixes, suffixes and stem are specified. The system provides POS data for each entry and gives the English gloss of the stem which allows the tool to act as a dictionary.

b) **Compatibility Tables:** Large set of allowable Arabic morphological patterns that are used to analyze the Arabic text and verify its validity.

c) **Analysis Engine:** The results of the other components are exploited to produce a number of analyses that are morpheme analyses (i.e. Buckwalter POS tag).

Arabic NER systems benefit from BAMA to extract morphological information to be used as features whether the NER system is rule-based or ML-based. The main morphological features extracted by BAMA are POS tag and Affixes. However, the English gloss may be utilized as a feature as well. Examples of Arabic NER systems that have used BAMA as a morphological analyzer to extract morphological features are Elsebai, Meziane and BelKredim (2009) and Farber et al. (2008).

2.6.3.2. MADA

MADA[7] stands for Morphological Analysis and Disambiguation for Arabic (MADA, 2012). MADA system is combined with TOKAN system (i.e. Tokenizer which applies different schemes) to form one package (i.e. MADA+TOKAN) which has the ability to perform a number of Arabic processing tasks including tokenization, diacritization (i.e. the use of short vowels instead of diacritics), morphological disambiguation, POS tagging, stemming and lemmatization (MADA, 2012). MADA is able to extract 14 morphological features (as illustrated in Table 2.5) for each data entry based on a SVM trained model (Habash et al., 2010). The system gives also the English gloss for the disambiguated entries.

The MADA's various morphological features can be exploited by rule-based or ML-based NER systems for Arabic in order to support the robustness of the rules and/or the learned models. MADA has been used in several research studies about NER for Arabic

[6] LDC Catalog No.: LDC2004L02, on http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2004L02
[7] Available for free download on http://www1.ccls.columbia.edu/MADA/MADA_download.html

such as Benajiba, Diab and Rosso (2008a; 2008b; 2009a; 2009b) and Farber et al. (2008). In this thesis, MADA is utilized as a morphological analyzer to extract the morphological features of each word in an input text and then use them as a part of the feature set.

No.	Feature	Feature Value Definition
1	Aspect	*Verb aspect:* Command, Imperfective, Perfective or Not applicable (NA)
2	Case	*Grammatical case:* Nominative, Accusative, Genitive, NA or Undefined
3	Gender	*Nominal gender:* Feminine, Masculine, or NA
4	Mood	*Grammatical Mood:* Indicative, Jussive, Subjunctive, NA or Undefined
5	Number	*Grammatical number:* Singular, Plural, Dual, NA or Undefined
6	Person	*Person information:* 1st , 2nd , 3rd or NA
7	State	*Grammatical state:* Indefinite, Definite, Construct/Poss/Idafa, NA or Undefined
8	Voice	*Verb voice:* Active, Passive, NA or Undefined
9	POS	*POS Definition:* Nouns, Number Words, Proper Nouns, Adjectives, Adverbs, Pronouns, Verbs, Particles, Prepositions, Abbreviations, Punctuation, Conjunctions, Interjections, Digital Numbers or Foreign/Latin
10	Proclitic 3	*Question proclitic:* No proclitic (NP), NA or Interrogative Particle >*a*
11	Proclitic 2	*Conjunction proclitic:* NP, NA, Conjunction *fa*, Response conditional *fa*, Subordinating conjunction *fa*, Conjunction *wa*, Particle *wa* or Subordinating conjunction *wa*
12	Proclitic 1	*Preposition proclitic:* NP, NA, Particle *bi*, Preposition *bi*, Preposition *ka*, Emphatic Particle *la*, Preposition *la*, Response conditional *la*, Jussive *li*, Preposition *li*, Future marker *sa*, Preposition *ta*, Particle *wa*, Preposition *wa*, Preposition *fy*, Negative particle *lA*, Negative particle *mA*, Vocative *yA*, Vocative *wA* or Vocative *hA*
13	Proclitic 0	*Article proclitic:* NP, NA, Determiner, Negative particle *lA*, Negative particle *mA*, Relative pronoun *mA* or Particle *mA*
14	Enclitics	*Pronominal:* No enclitic, NA, 1st person (plural|singular), 2nd person (dual|(feminine (plural|singular))|(masculine (plural|singular))), 3rd person (dual|(feminine (plural|singular))|(masculine (plural|singular))), Vocative particle, Negative particle *lA*, Interrogative pronoun (*ma*|*mA*|man), Relative pronoun (*ma*|*mA*|man) or Subordinating conjunction (*ma*|*mA*)

Table 2.5: MADA Morphological Features and their Description

2.6.3.3. AMIRA

AMIRA[8] is a free Arabic text processing toolkit based on SVM approach which is applied using Yamcha. Tokenization, POS tagging and Base Phrase Chunking (BPC) are the main functionalities of AMIRA (Diab, 2009). Deriving BPC is one of the distinctive characteristics of AMIRA in which BPC is the task of identifying syntactic phrases such as noun phrases

[8] Available for free on https://foliodirect.net/

(NPs) and verb phrases (VPs) within an Arabic text. BPC has proven its usefulness for Information Extraction systems (Diab, 2009). Arabic NER systems use AMIRA to derive BPC which gives indications to candidate phrases especially NPs that may contain named entities. A demo of the system is available on http://nlp.ldeo.columbia.edu/amira/. AMIRA has been used in a number of Arabic NER studies such as Benajiba, Diab and Rosso (2008a; 2008b; 2009a; 2009b).

2.6.3.4. Research and development international toolkit (RDI)

RDI[9] is a collection of Arabic processing tools including Diacritizer/Vowelizer, Morphological Analyzer, Part-of-Speech Tagger, Lexical Semantic Analyzer, Text Search Engine and Lexical Dictionaries. Arabic NLP systems of different tasks may use RDI toolkit to enable Arabic text processing. Arabic NER systems, rule-based or ML-based, may utilize RDI toolkit in extracting tokens' (i.e. words) features presented in an Arabic text.

Each of these tools has an online demo that can be used for free on the official website: http://www.rdi-eg.com/technologies/arabic_nlp.htm yet the tools are not available for free download. AbdelRahman et al. (2010) have used RDI toolkit to extract the morphological features of the input data to be considered by their proposed Arabic NER system.

The selection of which Arabic processing tool to be utilized in Arabic NER systems depends on the functions need to be fulfilled by the Arabic processing tool. From the description given earlier for each Arabic processing tool, a developer may select which tool to be utilized in building Arabic NER system according to the features and functions of each tool. For example, AMIRA is distinctive with its ability to derive BPC, while MADA is distinctive with its morphological disambiguation ability along with extensive morphological analysis capabilities.

2.7. Related work

This section illustrates the related work of the three types of NER systems including rule-based NER systems, ML-based NER systems and Hybrid NER systems.

[9] The Official website http://www.rdi-eg.com/

2.7.1. Rule-Based NER Systems

Maloney and Niv (1998) have presented TAGARAB system which is one of the early attempts to tackle NER for Arabic. TAGARAB is a rule-based system where a pattern matching engine is combined with a morphological tokenizer to recognize named entities of different categories including Person, Entity, Location, Number and Time. The stream of the system is as follows: firstly, the input dataset goes into the morphological tokenizer to extract tokens and relevant morphological features such as Part-of-speech tags and Nominal suffixes; secondly, the output of the morphological tokenizer goes as an input into the Name Finder (i.e. pattern matching engine) to detect and extract the named entities in order to produce an annotated version of the input dataset. The experimental results showed that the combination between the Name finder and the morphological tokenizer outperforms the individual Name finder in terms of accuracy. Time and Number entities did not show much difference in the results after the consideration of the morphological tokenizer in the recognition process.

Maynard et al. (2001) have introduced MUSE system which is a grammar-based NER system implemented in GATE framework. MUSE is composed of a tokenizer, gazetteers and grammars. Various kinds of text, such as emails and scientific articles, can be handled by MUSE for NER. After a preliminary analysis of the input text, the set of processing resources used by MUSE is manipulated to extract named entities taking into account the text type. The manipulation process is performed manually in prior of the recognition phase. The evaluation was conducted on English datasets to extract named entities of various types including Person, Date, Location, Organization, Time, Money, Percent, Email, URL, Telephone, IP and Identifier.

Riaz (2010) has introduced a rule-based system to tackle the task of Urdu NER. Although Urdu words are composed of Arabic letters and some words are common between the two languages, Arabic language processing systems would not work on Urdu texts and vice versa. Due to the lack of large annotated datasets, linguistic rules were constructed instead of training a statistical model to extract NEs of types: person, location, organization, date and number. The system outperforms the Conditional Random Fields based NER systems proposed in IJCNLP 2008 as reported by Riaz (2010).

Mesfar (2007) has developed an Arabic component under NooJ linguistic environment to enable Arabic text processing and NER. The component consists of a tokenizer, morphological analyzer and Arabic NER. The NE finder utilizes a set of predefined

dictionaries and indicators lists to support the grammar construction. The system identifies named entities of types: Person, Location, Organization, Currency, and Temporal expressions. Morphological information, which is provided by the morphological analyzer, is used by the system to extract unclassified proper nouns and thereby enhance the overall performance of the system.

Another work which adopts the rule-based approach for NER was done by Shaalan and Raza (2007). PERA is a grammar-based system which is built for identifying Person names in Arabic scripts with high degree of accuracy. PERA is composed of three components including gazetteers, grammars and filtration mechanism. Whitelists of complete Person names are provided to the gazetteer component in order to extract the matching names out of the input text as Person names regardless of the grammars. Afterwards, the input text is presented to the grammatical rules, which are in the form of regular expressions, to identify the rest of Person named entities. Finally, the filtration mechanism is applied on the entities detected and classified through certain grammatical rules in order to exclude invalid entities. PERA achieved satisfactory level of results when applied on ACE and Treebank Arabic datasets.

As a continuation of Shaalan and Raza (2007) research work, NERA system was introduced in Shaalan and Raza (2008; 2009). NERA is a rule-based system which is capable of recognizing named entities of 10 different types including Person, Location, Organization, Date, Time, ISBN, Price, Measurement, Phone Numbers and Filenames. The implementation of the system was in FAST ESP framework where the system has three components as the PERA system (i.e. gazetteers, grammars and filtration mechanism) with the same functionalities to cover the 10 named entities types. The Authors have constructed their own corpora from different resources in order to have a representative number of instances for each entity type.

Elsebai, Meziane and BelKredim (2009) have proposed a rule-based NER system that integrates pattern matching with morphological analysis to extract Person names from Arabic text. The pattern matching component utilizes lists of keywords in its structure without any gazetteers of Person names embedded within the system. The system is implemented in GATE environment where different processing resources are utilized such as tokenizer. The performance of the system was compared to the PERA performance despite of the fact that the PERA system is evaluated using different datasets than the ones used for Elsebai, Meziane and BelKredim (2009)'s system evaluation.

2.7.2. ML-Based NER Systems

Baluja, Mittal and Sukthankar (2000) have suggested the use of a ML-based approach to tackle the problem of NER in English scripts. The system consists of a tokenizer, gazetteers and decision tree classifier. The feature set is composed of 29 Boolean features distributed among three types of features including word level features such as capitalization, dictionary-based features such as whether the encountered word belongs to a dictionary, POS features such as verb and proper-noun, and punctuation features such as period and comma. A set of experiments has been conducted considering different combinations of features to find the feature set with the highest performance in terms of accuracy measures. The influence of taking contextual information into consideration during the learning process was studied in which the features of n words before and after the encountered word in the input data are added to the feature set of such word. According to the experimental results, the system achieves its highest performance when all types of features are considered in the feature set. The contextual information effect was studied when n = 0, 1, 2, 6 and the highest results were achieved when n = 1, 2 with a very small difference between the scores of the two n values.

Zhou and Su (2002) have developed a ML-based NER system to recognize different types of entities including named, time and numerical expressions from English text. The used ML approach is hidden Markov Model to integrate features of two sub-categories including internal evidence and external evidence. The internal evidence concerns about the word-level feature types including Deterministic features such as InitialCaps and OneDigitNum, Semantic features such as SuffixPercent (e.g. %) and SuffixTime (e.g. GMT), and Gazetteer features which determine the presence of the candidate word in one or more of the system gazetteers such as Person, Organization and Location gazetteers. On the other hand, the features representing the external evidence are the Contextual features that can be derived from the context to extract the relationships between previously recognized entities and the candidate word within the input text. The system was applied on MUC-6 and MUC-7 English shared tasks and the results showed that the performance reaches its highest rates when all the features contribute to the classification process.

Mayfield, McNamee and Piatko (2003) have proposed a language-independent statistical-based system for NER. The system exploits SVM approach to handle large amount of features with minimal overfitting. Examples of the features are word's length, POS tag, if the word is between double quote marks or not, and if a dash existed in the word or not. The features of a range of words surrounding the encountered word contribute to the learning and classification processes as well. The window size is seven

words centered by the targeted word. The system was applied on CoNLL2003 English and German datasets for training and evaluation.

Paliouras et al. (2000) have introduced a domain-specific NER system which utilizes C4.5 algorithm (i.e. decision trees) as the machine learning technique to detect and classify named entities of types Person and Organization. Each word in the input text is represented with two features: the word's POS tag and a gazetteer tag of which gazetteer the word belongs to if any. The classifier is fed with positive and negative examples of NE phrases. Each example is a sentence with maximum length of 10 words, and each sentence is represented with 28 features distributed as 2 features for each word in the sentence, 4 features for the two left adjacent words of the sentence as well as another 4 features for the two right adjacent words. Paliouras et al. (2000) have manually produced a set of grammatical rules to be compared with the statistical-based NER system in terms of performance, and the results showed that the decision tree-based system outperforms the manual rules. The system was evaluated using 10-fold cross validation to ensure having unbiased results.

MENERGI (Chieu and Ng, 2002) is a NER system based on maximum entropy approach. The system makes use of the global context to extract information from the entire document which will be fed to the system as features. There are two types of features in the MENERGI feature set including local features and global features. The local features are extracted based on the encountered word and its first immediate neighbors in both sides such as the case of the first letter and the lexical features of the word. On the other hand, the global features are derived from recognizing names and then encountering their abbreviated forms later in the same document, for example, if the name previously recognized is associated with an indicator or not (e.g. Person prefix: Mr.). The system was evaluated using MUC-6 and MUC-7 datasets, and the results were comparable to previous ML-based NER systems.

Benajiba, Rosso and Bened´ı (2007) have developed an Arabic NER system, ANERsys 1.0, which relies on Maximum Entropy (ME) technique in order to generate the classification model. The authors have built their own Arabic linguistic resources: ANERcorp (i.e. an annotated corpus) and ANERgazet (i.e. gazetteers) due the lack of free Arabic resources. The system can recognize four types of entities including person, location, organization and miscellaneous. ANERsys 1.0 system used to have difficulties with detecting NEs that are composed of more than one token; hence Benajiba, Rosso and Bened´ı (2007)'s research has been followed by another work, ANERsys 2.0 (Benajiba and Rosso, 2007), which adopts 2-step mechanism for NER: the first step is detecting the start and the end points of each NE, while the second step is classifying the detected NEs.

Benajiba and Rosso (2008) applied conditional random fields (CRF) approach instead of maximum entropy (ME) approach in their previous work, ANERsys 2.0, as an attempt to improve the performance. The feature set used in ANERsys 2.0 was used in this research to allow comparison. The features are POS tags and base phrase chunks (BPC), gazetteers (i.e. ANERgazet) and nationality. The CRF-based system outperformed the ME-based system in terms of precision, recall and F-measure.

Benajiba, Diab and Rosso (2008a) have developed another NER system based on Support Vector Machines (SVM) and applied different feature sets on the system to observe their impacts on the performance. In their previous work (Benajiba and Rosso, 2008), language specific features were not considered, and the system was not evaluated using standard corpora; hence they tried to overcome these issues in this research. The features considered in this work are contextual, lexical, Gazetteers, Morphological features, POS-tags and Base Phase chunks (BPC), nationality and corresponding English capitalization. SVM can handle noise in the text, and the availability of large set of features improves the generalization. The system has been evaluated using ACE Corpora and ANERcorp. The proposed system achieved the best results in terms of the standard NER measures when all the features are included in the feature set. According to the experimental analysis, the consideration of both language independent and language specific features contributes positively on the performance of Arabic NER task.

A simplified feature set has been proposed by Abdul-Hamid and Darwish (2010) to be considered in Arabic NER task. The feature set was used in CRF-based Arabic NER system to recognize three types of named entities including Person, Location and Organization. Abdul-Hamid and Darwish (2010) avoided the use of any gazetteers, morphological or syntactic features in attempt to prove that surface features can be used effectively for Arabic NER task. The suggested features are leading and trailing character n-gram, word position, word length, word unigram probability, the preceding and succeeding words n-gram and character n-gram probability. The results showed that their proposed system outperformed the CRF-based NER system of Benajiba and Rosso (2008).

In another study, Benajiba, Diab and Rosso (2008b) have investigated the sensitivity of each entity type to different sets of features (i.e. the same features used by Benajiba, Diab and Rosso (2008a)) and in order to do that they built different classifiers for each entity type adopting SVM and CRF approaches. The features were studied in isolation and in gradual combinations. Three standard data sets (i.e. ACE 2003, 2004 and 2005) were used to evaluate the performance of the classifiers. According to the Empirical results, it cannot be stated that CRF is better than SVM or vice versa for Arabic named entity recognition, each type of entities is sensitive to different features and each feature plays a role in

recognizing the NE in different degrees. Also, further studies (Benajiba, Diab and Rosso, 2009a; 2009b) have supported these findings and confirmed the importance of considering language independent and language specific features in Arabic NER systems.

AbdelRahman et al. (2010) have introduced an integration of two machine learning approaches in order to handle Arabic NER task including CRF (i.e. a supervised technique) and bootstrapping pattern recognition (i.e. a semi-supervised technique). The feature set used with the CRF classifier includes word-level features, POS tag, Base Phase Chunks (BPC), gazetteers and morphological features. The system is developed to extract NEs of 10 types including Person, Location, Organization, Job, Device, Car, Cell Phone, Currency, Date and Time. The performance of the system was compared to the performance of LingPipe tool which adopts HMM chunker over the ANERcorp corpus. The results showed that the proposed system outperformed the LingPipe NER system.

Farber et al. (2008) have suggested the integration of morphological tagger with Arabic NER system. The authors claim that the morphological information produced by a morphological tagger is significant for Arabic NER. The proposed strategy was to utilize the output of a morphological tagger as features of the input text. These features will be used by the learner (i.e. the classifier) in order to identify and classify named entities of types Person, Organization, and Geo-political entities (GPEs). The system adopts the structured perceptron approach for Arabic NER. The empirical results show that the morphological features have improved the performance of NER system for Arabic.

2.7.3. Hybrid NER Systems

Srihari, Niu and Li (2000) have proposed a hybrid NER system which exploits two ML approaches and pattern matching rules to extract named entities of several types including Person, Location, Organization, Date, Money, Time and Percentage. The system is composed of four main modules. The first module is the pattern matching rules used to recognize temporal expressions and numerical expressions. The second module is a Maximum Entropy model built to utilize the gazetteers and the contextual information in order to provide a preliminary recognition of Person and Location entities. The third module is the HMM classifier which gives the final tagging for the entities of types: Person, Location, and Organization. The final module is another Maximum Entropy produced to identify sub-categories such as Government and Airport. The system is evaluated using MUC-7 dataset and the results show that using the gazetteer module improves the performance of the NER system.

Seon et al. (2001) have introduced a hybrid approach to attack the problem of NER for Korean. The ML approaches used by the system are Maximum Entropy and Neural

Networks. The Maximum Entropy is utilized to resolve the problem of unknown words that do not exist in any of the predefined dictionaries within the system, while Neural Networks use lexical information to deal with ambiguity when a duplicate tag is found. At the last stage, the pattern-selection rules are exploited to combine adjacent words into a named entity. The proposed system is developed to extract NEs of types: Person, Location and Organization.

Mencius is a NER system for Chinese proposed by Tsai et al. (2004). The rule-based and ML-based methods are combined in Mencius in order to enhance the recognition capabilities for Person, Location, and Organization entity types. The output of the rule-based module is used as an input in the form of features for ML-based module which is a Maximum Entropy Model. The feature set consists of internal (i.e. category-dependent) features utilized to distinct between different named entity categories. The authors have built their own corpus to evaluate the performance of Mencius. The evaluation was conducted using 10-fold cross validation where different settings were considered. The highest results are achieved when the hybrid system is used along with word tokenization.

Ferreira, Balsa and Branco (2007) have presented a hybrid NER system for Portuguese. A rule-based module is utilized to detect and classify numbers, addresses, measures and time. On the other hand, a hybrid module is used to deal with a group of entity types including Person, Location, Organization, Work (e.g. movies and books), Event, and Miscellaneous. Some word-level features need to be provided as lexical information to the rule-based module including POS tag, Lemma tag and inflection feature which lead to correctly invoke of the defined rules. The hybrid module is composed of a statistical tagger and a rule-based application. Two statistical approaches were studied, HMM and Maximum Entropy algorithms. The rule-based application is used for error correction after the input text is tagged by the statistical tagger as a post processing step. The experimental results show that the Maximum Entropy-based tagger outperforms the HMM-based tagger, and that the utilization of a rule-based system as a post processing step improves the performance of the system.

Nguyen and Cao (2008) have proposed a hybrid NER system which deals with ambiguity in an incremental approach. The system is designed to identify Person, Location and Organization named entities. The recognition of NEs within a text involves two main steps. The first step is to use a rule-based module to select NE candidates, and the second step is to rank those candidates and identify named entities using a machine learning model. This strategy is applied incrementally to filter the candidates in an input text and solve ambiguity through exploiting the features of previously identified named entities in

each round for next rounds and so forth. The authors claim that this approach is language-independent and can be utilized as a disambiguation technique for different language.

Petasis et al. (2001) have introduced a hybrid method to facilitate maintaining rule-based NER systems. The method is to use a ML-based system to evaluate the performance of the rule-based NER system, and then maintaining the rule-based NER system when it is needed. The ML-based NER system uses the annotated output of the rule-based system in the training phase; hence manually tagged data set is not required since it is automatically available by the rule-based system. Then, the two systems are applied on a new data set and their results are compared. The disagreements will indicate the need of updating the rule-based system. The ML technique adopted by this work is C4.5 (i.e. decision trees). The proposed method is evaluated with two rule-based NER systems, a Greek system and a French system. The systems can recognize named entities of types: person, location and organization. The experimental analysis shows that this method can help deciding when to maintain a rule-based NER system.

To the best of our knowledge, the only Arabic NER system which has been developed based on hybrid approach is by Abdallah, Shaalan and Shoaib (2012). The hybrid system is composed of two main components including a rule-based system and ML-based system in order to identity Person, Location and Organization named entities within Arabic scripts. The rule-based component is implemented in GATE framework based on the technical reports of the NERA system which was developed by Shaalan and Raza (2008; 2009). On the other hand, the ML-based component utilizes decision trees approach to build the classifier. Each word is represented with a vector of features including Rule-based features and Machine learning features. The rule-based features are derived from the rule-based system's annotations on the input text where also the annotations of the immediate two neighbors on both sides of each word are considered as features of this category. While the machine learning features are the word's length, POS tag, Noun flag (i.e. whether the POS tag is Noun or not), gazetteers features, statement-end (i.e. check the dot existence on both sides of the word), Prefix features and suffix features. The system is evaluated using ACE 2003 and ANERcorp. The experimental results show that the hybrid system outperforms the CRF-based NER system built by Benajiba and Rosso (2008) as reported by Abdallah, Shaalan and Shoaib (2012) in terms of performance on ANERcorp dataset.

2.8. Conclusion

Recently, Named Entity Recognition (NER) has received a lot of attention in the field of NLP research for various languages, such as English, Germen, Chinese and Arabic, due to the important role played by NER in different NLP systems which leads to optimizing the overall performance of those systems. A NER system may be rule-based, ML-based or hybrid system. Rule-based NER systems rely on handcrafted grammatical rules written by linguists and require tagged corpora only for evaluation. Therefore, any maintenance or update applied to rule-based systems is labor and time consuming especially if the linguists are not available. On the other hand, ML-based NER systems require the availability of tagged corpora for training and testing phases. ML-based NER systems are adoptable and updatable with minimal time and effort in which linguists are not needed since no handcrafted rules are utilized within the systems. The feature set employed by the ML approach has a significant impact on the performance of the ML-based NER system. Hybrid NER systems combine the handcrafted grammatical rules with the statistical approaches in order to improve the performance of NER systems. Therefore, the decision of which NER approach to be adopted depends mainly on the resources intended to be used; for a ML-based NER system, a feature set needs to be selected and linguistic resources including annotated corpora and gazetteers need to be available, while for a rule-based NER system, specialists in the natural language of the system are required in order to manually acquire grammatical rules. NER for Arabic is in its early stages where opportunities for improvement in the performance still available. Building an Arabic NER system requires tagged corpora whether for training and testing or only testing, Gazetteers/trigger words lists, and a suitable feature set in the case ML-based and hybrid systems. To the best of our knowledge, only one hybrid NER system for Arabic has been introduced by Abdallah, Shaalan and Shoaib (2012) to recognize Person, Location and Organization named entities. Therefore, the field of hybrid NER for Arabic needs further investigations and studies to enhance the scope and improve the overall performance. In this thesis, we contribute to the field of hybrid NER for Arabic in which a hybrid NER for Arabic is proposed to handle the recognition of 11 types of named entities including Person, Location, Organization, Date, Time, Price, Percent, Measurement, Phone Number, ISBN and File Name via implementing a rule-based system to be integrated with a ML-based system to form the hybrid system.

Chapter 3

Data Collection

This chapter describes the process of data collection. The linguistic resources used in this research are listed and discussed according to the category (i.e. Corpora or Gazetteers). The preprocessing phase of the datasets is described as well.

Various linguistic resources are necessary for our research in order to enable Named Entity Recognition task for Arabic with scope of eleven categories of named entities including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name. As mentioned in section 2.4, the linguistic resources are of two main categories including corpora (i.e. datasets) and gazetteers (i.e. dictionaries). In this research, annotated corpora are required for the evaluation of the rule-based component and for the training and testing of the ML-based component. The collected gazetteers are utilized within the implemented rules and also within the feature set as the dictionary-based features.

3.1. Training and Testing Corpora

The datasets used in this research are ACE corpora, Arabic Treebank (ATB) Part1 v 2.0 dataset, ANERcorp, and our own corpus. ACE corpora and ATB dataset are available under license agreement[1], while ANERcorp[2] is freely available for research purposes. We also have built our own corpus to train and evaluate our system when it comes to identifying certain types of named entities.

[1] Available for BUiD under license from LDC
[2] Available to download on http://www1.ccls.columbia.edu/~ybenajiba/downloads.html

3.1.1. ACE Corpora

ACE[3] (i.e. Automatic Content Extraction) is a project established on 1990 with a main objective which is to enhance text processing in order to support information extraction. The extracted information can be of different forms including events, relations and entities. ACE datasets are not for free yet can be obtained under license distributed by Linguistic Data Consortium[4] (LDC). ACE corpus comes with two types of files in in Standard Generalized Markup Language: data files (i.e. the raw text) and corresponding annotations files (i.e. entity information). ACE has three multilingual training corpora which include Arabic datasets as part of their contents:

- **ACE 2003 Multilingual Training Data**

  ACE 2003 dataset (Mitchell et al., 2003) is available under license from LDC with catalog number LDC2004T09 and ISBN 1-58563-292-9. The training data is distributed among two genres including Newswire (NW) and Broadcast News (BN).

- **ACE 2004 Multilingual Training Corpus**

  ACE 2004 dataset (Mitchell et al., 2005) is available under license from LDC with catalog number LDC2005T09 and ISBN 1-58563-334-8. The training data is distributed among three genres including Newswire (NW), Broadcast News (BN) and Arabic Treebank (ATB).

- **ACE 2005 Multilingual Training Corpus**

  ACE 2005 dataset (Walker et al., 2006) is available under license from LDC with catalog number LDC2006T06 and ISBN 1-58563-376-3. The training data is distributed among three genres including Newswire (NW), Broadcast News (BN) and Weblog (WL).

The entity information files of ACE 2003 corpus contain annotations for several types of named entities including Person, Facility, Organization and GPE (i.e. geographical and political entities). Later in ACE 2004 & 2005, another two categories have been added to ACE 2003 tag: vehicles and weapons. ACE 2005's entity information files have annotations for temporal expressions, which follow TIMEX2 specifications, and numerical expressions

[3] http://www.itl.nist.gov/iad/mig/tests/ace/ and http://projects.ldc.upenn.edu/ace/
[4] http://www.ldc.upenn.edu/

including money, phone number and percentage. Figure 3.1 and Figure 3.2 illustrate the format of ACE data files and Entity information files respectively. Recall, Figure 2.1 in the previous chapter illustrates a sample of Entity information files in ACE 2005.

In this research, NW and BN files are the ones targeted of each ACE corpus.

```
<DOC>
<DOCNO> AFA20001018.0000.0020 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 2000-10-18 07:26:00 </DATE_TIME>
<HEADER>
ار0022 4 ش 0088 قبر /افب-ضغص58 روسيا/غواصة
</HEADER>
<BODY>
<HEADLINE>
ارجاء انتشال جثث بحارة الغواصة
النووية الروسية الى الاسبوع المقبل
</HEADLINE>
<TEXT>
<P>
موسكو 10-18 (اف ب)- اعلن نائب رئيس
الوزراء الروسي ايليا كليبانوف اليوم
الاربعاء ان عملية انتشال جثث بحارة
الغواصة الروسية النووية كورسك ارجئت
الى الاسبوع المقبل بسبب الاحوال
الجوية السيئة في منطقة الكارثة.
</P>
<P>
وكان من المقرر ان تبدا العملية اليوم
الاربعاء.
</P>
<P>
وكانت الغواصة وعلى متنها 118 بحارا
غرقت في بحر بارينتس فى المنطقة
القطبية الشمالية في 12 اب/اغسطس الماضي
بسبب انفجار لم تتضح بعد اسبابه بشكل
اكيد.
</P>
</TEXT>
</BODY>
<TRAILER>
35 فلفل/ع ق/اغ موا _____
افب
180726 00 جمت اوك
</TRAILER>
</DOC>
```

Figure 3.1: Sample of an ACE 2003 Data File

```
▼<source_file URI="AFA20001018.0000.0020.sgm" SOURCE="newswire"
  TYPE="text" VERSION="2.0" AUTHOR="LDC" ENCODING="UTF-8">
 ▼<document DOCID="AFA20001018.0000.0020">
  ▼<entity ID="AFA20001018.0000.0020-E1">
     <entity_type GENERIC="FALSE">GPE</entity_type>
   ▼<entity_mention ID="1-1" TYPE="NAME" ROLE="LOC">
    ▼<extent>
     ▼<charseq>
        <!-- string = "موسكو" -->
        <start>179</start>
        <end>183</end>
       </charseq>
      </extent>
    ▼<head>
     ▼<charseq>
        <!-- string = "موسكو" -->
        <start>179</start>
        <end>183</end>
       </charseq>
      </head>
    </entity_mention>
   ▼<entity_attributes>
    ▼<name>
     ▼<charseq>
        <!-- string = "موسكو" -->
        <start>179</start>
        <end>183</end>
       </charseq>
      </name>
    </entity_attributes>
   </entity>
  </document>
 </source_file>
```

Figure 3.2: Sample of an ACE 2003 Entity Information file (Annotations file)

3.1.2. ATB Part1 v. 2.0 Dataset

Arabic Treebank Part1 v. 2.0 dataset (Maamouri et al., 2003) is available under license from LDC with catalog number LDC2003T06 and ISBN 1-58563-261-9. ATB dataset comes with two types of files including data files and annotation files. Each data file in ATB has a corresponding annotation file (available in XML and TXT formats) which contains POS tags for each word in the data file in order to support POS tagging task. However, ATB dataset is not originally produced for NER task; hence no annotations of named entities that appear in the dataset are available. Therefore in this research, ATB dataset has been manually annotated with named entity tags of certain categories to support NER task for Arabic. Figure 3.3 shows a sample of ATB data files:

```
<DOC>
<DOCNO>20000715_AFP_ARB.0003</DOCNO>
<HEADER>
ار0021 4 ر 0112 قبر /افب-ثجغ04 سيارات/فورمولا
</HEADER>
<BODY>
<HEADLINE>
جائزة النمسا الكبرى: انسحاب الايرلندي ايرفاين
</HEADLINE>
<TEXT>
<P>
 سبيلبرغ (النمسا) 15-7 (أ ف ب)- اعلن السائق الايرلندي ايدي ايرفاين
(جاغوار) انسحابه من سباق جائزة النمسا الكبرى، المرحلة العاشرة من
بطولة العالم لسباقات الفورمولا واحد (الفئة الاولى)، التي تقام غدا الاحد
على حلبة سبيلبرغ.
</P>
<P>
 وكان ايرفاين، الذي حل في المركز الاول في جائزة النمسا العام الماضي
على سيارة فيراري، شعر بآلام في بطنه اضطرته الى الانسحاب من التجارب، وهو
سيعود الى لندن لاجراء الفحوصات الضرورية حسب ما اشار فريق جاغوار.
</P>
<P>
وسيحل سائق التجارب في جاغوار البرازيلي لوسيانو بورتي مكان ايرفاين في
السباق غدا الاحد، الذي سيكون اولى خطواته في عالم سباقات الفورمولا واحد.
</P>
</TEXT>
<FOOTER>
 ب ب/ميش/م م فسب افب  ـــــــــــــــــــــــــــــــ
</FOOTER>
</BODY>
<TRAILER>
150756 00 جمت يول
</TRAILER>
</DOC>
```

Figure 3.3: Sample of an ATB Data File

3.1.3. ANERcorp Dataset

ANERcorp is an annotated dataset built by Yassine Benajiba for Arabic NER (Benajiba, Rosso and Bened´I, 2007). The size of the corpus is more than 150K words labeled to recognize named entities of four classes including Person, Location, Organization and Miscellaneous classes. The corpus follows the CoNLL tag set and IOB scheme (as described in section 2.3).

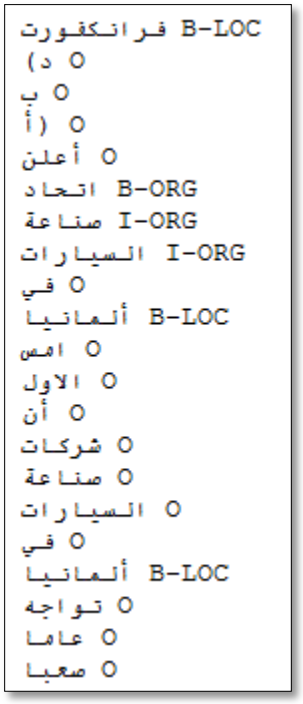

Figure 3.4: Sample of ANERcorp Dataset

3.1.4. Our Own Corpus

The previous listed datasets are not suitable for training and/or testing an Arabic NER system for named entities of types Phone Number, ISBN and File Name because of the insufficient number of NEs of these types within the available Arabic datasets. In order to have a dataset with a respective number of named entities of certain types including Phone Number, ISBN and File Name, we have prepared our own corpus using different internet resources as listed in Table 3.1. The corpus contains 126 phone numbers, 136 ISBNs and 160 file names in which the annotation process was manual. Figure 3.5 illustrate an annotated sample of the corpus we have built.

Named Entity Type	Internet Resources
Phone Number	http://arabicorpus.byu.edu/
ISBN	http://arabicorpus.byu.edu/ http://www.goethe.de/kue/lit/prj/kju/arindex.htm http://ar.wikipedia.org/wiki/ http://www.arableaguetunis.org/fr/biblio.htm http://www.souqalarab.com/ http://www.kfnl.org.sa/?action=showSection&id=25 http://www.kisr.edu.kw/Default.aspx?pageId=304 http://aamoudi.kau.edu.sa/ http://the-light-group.ahlamontada.com/t1874-topic
File Name	http://www.sh4arab.com/vb/archive/index.php/t-31349.html http://forums.cgway.net/cg44411/ http://support.microsoft.com/kb/950505/ar http://soft.sptechs.com/ http://www.aloots.com/modules http://www.felqalb.com/vb/ http://infomag.news.sy/ http://www.abdta.com/Courses/MSWindowsXP/ http://www.om-eddonia.com/vb/archive/index.php?t-44391.html http://earbab.mam9.com/t123-topic

Table 3.1: The internet resources used to build our own corpus

وفي ما قد يطرأ من تبدلات على هذا الوضع اذا تغيّرت الحكومة البريطانية. وعلى
هذا المرء ان يُقدم على القيام بما يجب قبل نهاية العام المالي الجاري. فالوقت
قصير ويجب القيام بعمل الآن. لمزيد من المعلومات ديفيد فريمان على رقم الهاتف
<PhoneNumber>5835333-0171</PhoneNumber>
وعلى رقم الفاكس
<PhoneNumber>3530743-0171</PhoneNumber>
ويتعين على المرء المعني ان ينظر في الوقت الراهن في عدد كبير من المسائل
المتعلقة بالشأن الضريبي. لكن اقتراب موعد الانتخابات البرلمانية يجعل هذه
الخطوة امراً ملحاً مستعجلاً. ويتعين على كل من له ممتلكات لا يستهان بها (حتى
بالنسبة الى غير المقيمين في المملكة المتحدة، من وجهة نظر النظام الضريبي في
الوقت الراهن، لكن الذين لهم ارتباطات مع المملكة) ان ينظر في وضعه الضريبي
وفي ما قد يطرأ من تبدلات على هذا الوضع اذا تغيّرت الحكومة البريطانية. وعلى
هذا المرء ان يُقدم على القيام بما يجب قبل نهاية العام المالي الجاري. فالوقت
قصير ويجب القيام بعمل الآن. لمزيد من المعلومات ديفيد فريمان على رقم الهاتف
<PhoneNumber>5835333-0171</PhoneNumber>
وعلى رقم الفاكس
<PhoneNumber>3530743-0171</PhoneNumber>
وتشحن ديوان الآن الاصدار السابع من برنامج الناشر الصحفي وسمعنا من الشركة ان
وضعها المالي الى تحسن مستمر وانها تمكنت من سداد كل ديوانها وتلقى برامجها
رواجاً طيبا. «ديوان» على رقم الهاتف
<PhoneNumber>5333 252 -1710</PhoneNumber>

Figure 3.5: Sample of our own corpus

3.2. The System's Gazetteers

The rule-based NER system in this research is built upon the technical reports of NERA system developed by (Shaalan and Raza, 2008). The implemented rules utilize a different set of gazetteers/dictionaries for each category of named entities. The gazetteers for Person, Location and Organization names extractors were collected in a related work by (Abdallah, Shaalan and Shoaib, 2012) where the rules and the gazetteers are also based on the technical reports of NERA system. The gazetteer sets for the rest of the extractors (i.e. for eight NE types including Data, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name) are prepared as part of this research. For data collection of the gazetteers, the samples found in the NERA technical reports gave us guidelines and indications of the valid contents to be collected for each gazetteer. Various World Wide Web resources were helpful in the data collection of the gazetteers.

3.2.1. Gazetteers for Person, Location and Organization Extractors

The gazetteers, i.e. dictionaries, prepared for the Person names extractor are listed in Table 3.2. Examples for the contents of each gazetteer mentioned in Table 3.2 are illustrated in Table 3.3 with transliteration[5] and English translation.

Gazetteer name	Description
Complete Names	List of complete Person names (acts as a Whitelist)
First Names	List of Person names which represent First names
Middle Names	List of Person names which represent Middle names
Last Names	List of Person names which represent Last names
Honorifics	List of Person names' Honorifics
Person Titles	List of Person Titles that may be associated to Person names
Job Titles	List of Job Titles that may be associated to Person names
Location	List of Location names that may be associated to Person names
Numbers	List of Ordinal numbers which usually appear with Honorifics
Person Indicators	List of Person Indicators that may appear before or after Person names
Laqabs	List of Persons' Laqabs (i.e. nickname or surname)

Table 3.2: The gazetteers prepared for the Person names extractor

[5] Habash-Soudi-Buckwalter transliteration scheme

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Complete Names	أحمد ياسين		محمود درويش	
	Âahmad yaAsiyn	Ahmed Yassin	mahmud darwiyš	Mahmoud Darwish
First Names	زايد		محمود	
	zaAyid	Zayed	mahmud	Mahmoud
Middle Names	سلطان		محمد	
	sultaAn	Sultan	muham~ad	Mohammed
Last Names	نهيان		درويش	
	nahyaAn	Nahyan	darwiyš	Darwish
Honorifics	الرئيس		الملك	
	Alraŷiys	President	Almalik	King
Person Titles	الشيخ		السيد	
	Alšiyx	Sheikh	Alsay~id	Mr.
Job Titles	الجراح		الرسام	
	Aljar~aAh	Surgeon	Alras~aAm	Painter
Location	اليابان		أثينا	
	AlyaAbaAn	Japan	ÂaΘiynaA	Athens
Numbers	الأول		الرابع	
	Alaw~al	First	AlraAbiç	Fourth
Person Indicators	رئيس الوزراء		المشرف الرياضي	
	raŷiys AlwuzaraA'	Prime minister	Almušrif AlriyaAdiy	Sports Supervisor
Laqabs	الأمين		الأزهر	
	AlÂmiyn	Custodian	AlÂzhar	Al-Azhar

Table 3.3: Examples of entries for each Person names gazetteer

The gazetteers prepared for the Location names extractor are listed in Table 3.4. Examples for the contents of each gazetteer mentioned in Table 3.4 are illustrated in Table 3.5 with transliteration and English translation.

Gazetteer name	Description
Direction1	List of directions (i.e. in the primitive form)
Direction2	List of directions (i.e. in the definite form)
Direction3	List of directions (i.e. in the primitive form with suffix "ي")
Direction4	List of directions (i.e. in the definite form with suffix "ي")
Direction5	List of directions (i.e. in the definite form with suffix "ية")
City Names	List of cities
Country Names	List of countries
State Names	List of states
Capital Names	List of Capitals
Administrative Divisions	List of administrative divisions

Country Preceding Indicators	Lists of words that may appear before country names
Country Post Indicators	Lists of words that may appear after country names
City Preceding Indicators	Lists of words that may appear before city names
City Post Indicators	Lists of words that may appear after city names
Continents	List of continents' names
Monuments	List of monuments' names
Mountains	List of mountains' names
Rivers	List of rivers' names
Places	List of places' names
Oceans, Seas and Islands	List of oceans, seas and islands' names

Table 3.4: The gazetteers prepared for the Location names extractor

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Direction1	غرب		شرق	
	γarb	West	šarq	East
Direction2	الغرب		الشرق	
	Alγarb	The West	Alšarq	The East
Direction3	غربي		شرقي	
	γarbiy	In the West of	šarqiy	In the East of
Direction4	الغربي		الشرقي	
	Alγarbiy	The Western	Alšarqiy	The Eastern
Direction5	الغربية		الشرقية	
	Alγarbiy~aħ	The Western	Alšarqiy~aħ	The Eastern
City Names	القدس		طوكيو	
	Alquds	Jerusalem	Tuwkyuw	Tokyo
Country Names	فلسطين		إسبانيا	
	filisTiyn	Palestine	ĂsbaAnyaA	Spain
State Names	القاهرة		نيويورك	
	AlqaAhiraħ	Cairo	nyuwyuwrk	New York
Capital Names	بيروت		دمشق	
	bayruwt	Beirut	dimašq	Damascus
Administrative Divisions	جمهورية		مملكة	
	Jumhuwriy~aħ	Republic	mamlakaħ	Kingdom
Country Preceding Indicators	مملكة		جمهورية	
	mamlakaħ	Kingdom	Jumhuwriy~aħ	Republic
Country Post Indicators	المتحدة		الديمقراطية	
	Almut~ahidaħ	United	AldiymuqraATiy~aħ	Democratic
City Preceding Indicators	مدينة		مطار	
	madiynaħ	City	maTaAr	Airport

City Post Indicators	العاصمة		عاصمة المالية	
	AlçaASimaħ	The Capital	çaASimaħ AlmaAliy~aħ	Financial Capital
Continents	أفريقيا		أستراليا	
	ÂafriyqyaA	Africa	ÂustraAlyaA	Australia
Monuments	قوس النصر		بيكادلي	
	Qaws AlnaSr	Triumphal Arch	biykaAdiliy	Piccadilly
Mountains	جبال الألب		جبال الهملايا	
	jibaAl AlÂlb	Alps	jibaAl AlhimilaAyaA	Himalayas
Rivers	نهر النيل		نهر التايمز	
	nahr Alniyl	The Nile	nahr AltaAymz	River Thames
Places	ميدان		شارع	
	maydaAn	Field	šaAriç	Street
Oceans, Seas and Islands	المحيط الهادي		البحر الاسود	
	AlmuhiyT AlhaAdiy	Pacific Ocean	Albahr AlÂswad	Black Sea

Table 3.5: Examples of entries for each Location names gazetteer

The gazetteers prepared for the Organization names extractor are listed in Table 3.6. Examples for the contents of each gazetteer mentioned in Table 3.6 are illustrated in Table 3.7 with transliteration and English translation.

Gazetteer name	Description
Complete Company Names	List of complete Organization names (acts as a Whitelist)
Business Types	List of business/company types or structures
Company Following Indicator	List of Organization suffix words (not part of the name)
Company Following Known Part	List of Organization suffix words (as part of the name)
Company Preceding Indicator	List of Organization prefix words (not part of the name)
Company Preceding Known Part	List of Organization prefix words (as part of the name)
Locations	List of Location names
Prefix Business	List of words that may appear prior to business types

Table 3.6: The gazetteers prepared for the Organization names extractor

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Complete Company Names	سوني إركسون		مايكروسوفت	
	suwniy Ăiriksuwn	Sony Ericsson	maAykruwsuwft	Microsoft
Business Types	خدمات الانترنت		الخدمات الطبية	
	kadamaAt AlAintirnit	Internet Services	AlkadamaAt Altiby~aħ	Medical Services
Company Following Indicator	ش م م		وشركاه	
	Š m m	LLC	wašurakaAh	and Co. (partners)

Company Following Known Part	انترناشيونال		للأخبار	
	AintirnaAšyuwnaAl	International	lilÂaxbaAr	News
Company Preceding Indicator	بورصة		صحيفتي	
	buwrSaħ	Stock Exchange	Sahiyfatiy	Newspaper
Company Preceding Known Part	شركة		بنك	
	šarikaħ	Company	bank	Bank
Locations	أثيوبيا		أريزونا	
	ÂaΘyuwbyaA	Ethiopia	ÂariyzuwnaA	Arizona
Prefix Business	التجارية		الزراعية	
	AltijaAriy~aħ	Commercial	AlzraAçy~aħ	Agricultural

Table 3.7: Examples of entries for each Organization names gazetteer

3.2.2. Gazetteers for Date, Time, Price, Measurement & Percent Extractors

The gazetteers built for the **Date NE extractor** are listed in Table 3.8 with the description and size in words for each gazetteer. The gazetteers in bold font were not originally built in NERA yet we have built them to facilitate the implementation of the rules and enhance the rules as well. Examples for the contents of each gazetteer mentioned in Table 3.8 are illustrated in Table 3.9 with transliteration and English translation.

Gazetteer name	Description	Size
Month names	List of Month names including the various spelling variations and the different calendar schemes such as Gregorian and Hijri.	372
Weekdays	List of the days in a week in literal form including the various spelling variations	22
Relative Word	List of words that are considered as part of dates in Arabic, including the various spelling variations, such as "الجاري" and "الحالي" (the current)	24
Month's days	List of days in a month (1-31) in literal form including the various spelling variations	166
Hundreds	List of numbers for hundreds in literal form including the various spelling variations	36
Tens	List of numbers for tens in literal form including the various spelling variations	40
Units	List of numbers for units in literal form including the various spelling variations	75

Thousands	List of numbers for thousands in literal form including the various spelling variations	31
Year Type	List of year types including the various spelling variations e.g. "هجرية" (Hijri)	17
Year Range	List of words that give indications for year ranges including the various spelling variations e.g. "حتـى نهايـة" (until the end of)	8
Month Figures	List of months in numerical form using Arabic digits, e.g. 1, 2 and 3, and Indic digits e.g. ١, ٢ and ٣	42
Day Figures	List of days of month in numerical form using Arabic digits, e.g. 1, 2 and 3, and Indic digits e.g. ١, ٢ and ٣	80

Table 3.8: The gazetteers prepared for the Date extractor

Gazetteer name	**Examples of Entries**			
	Entry		**Entry**	
	Transliteration	**Translation**	**Transliteration**	**Translation**
Month names	ربيع الأول		يوليو	
	rabiyç AlÂw~al	Rabi' Al-awwal	yuwlyuw	July
Weekdays	الثلاثاء		الخميس	
	AlΘulaAΘaA'	Tuesday	Alxamiys	Thursday
Relative Word	الماضي		المقبل	
	AlmaADiy	Last	Almuqbil	Next
Month's days	الحادي عشر		السابع	
	AlHaAdiy çašar	The Eleventh	AlsaAbiç	The Seventh
Hundreds	تسعمائة		سبعمائة	
	tisçumaAŷaħ	Nine hundred	sabçumaAŷaħ	Seven hundred
Tens	عشرون		تسعون	
	çušruwn	Twenty	tisçuwn	Ninety
Units	أربعة		ثمانية	
	Âarbaçaħ	Four	ΘamaAniyaħ	Eight
Thousands	ألف		ألفين	
	Âalf	One thousand	Âalfayn	Two thousand
Year Type	ميلادي		هجري	
	miylaAdiy	Gregorian	hijriy	Hijri
Year Range	حتى نهاية		وإلى نهاية	
	Hat~aý nihaAyaħ	Until the end of	waĂilaý nihaAyaħ	And to the end of
Month Figures	١٢		12	
	12	12	12	12
Day Figures	٣		3	
	3	3	3	3

Table 3.9: Examples of entries for each Date gazetteer

The gazetteers built for the **Time NE extractor** are listed in Table 3.10 with the description and size in words for each gazetteer. The gazetteers in bold font were not originally built in NERA yet we have built them to facilitate the implementation of the rules and enhance the rules as well. Examples for the contents of each gazetteer mentioned in Table 3.10 are illustrated in Table 3.11 with transliteration and English translation.

Gazetteer name	Description	Size
Time Words	List of words that describe Time including the various spelling variations, such as "صباحًا" (morning)	41
Time Zones	List of time zones including the various spelling variation	40
Keyword Prefixes	List of words that may appear before Time words including the various spelling variations	10
Sharp Time	List of words that indicate an exact Time including the various spelling variations	11
Time Fractions	List of fractions that can be used to describe Time including the various spelling variations	18
Tens	List of numbers for tens in literal form including the various spelling variations	30
Hours	List of hours in literal form including the various spelling variations	58
Minutes	List of minutes in literal form including the various spelling variations (The same list is also utilized to represent Seconds)	162
Hour Figures	List of hours in numerical form using Arabic digits, e.g. 1, 2 and 3, and Indic digits e.g. ١, ٢ and ٣	70
Minute Figures	List of minutes in numerical form using Arabic digits, e.g. 1, 2 and 3, and Indic digits e.g. ١, ٢ and ٣ (The same list is also utilized to represent Seconds figures)	140

Table 3.10: The gazetteers prepared for the Time extractor

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Time Words	مساء		صباح	
	masaA'	Evening	SabaAH	Morning
Time Zones	توقيت غرينيتش		التوقيت المحلي	
	tawqiyt γriyniytš	Greenwich mean time (GMT)	Altawqiyt AlmaHaliy	Local Time
Keyword Prefixes	قبل		بعد	
	qabl	Before	baçd	After
Sharp Time	تماماً		بالضبط	
	tamaAmAã	Sharp	biAlDabt	Exactly
Time Fractions	ربع		نصف	
	rubç	Quarter	niSf	Half
Tens	عشرون		تسعون	
	çušruwn	Twenty	tisçuwn	Ninety
Hours	الخامسة		السابعة	
	AlxaAmisaħ	Five	AlsaAbiçaħ	Seven
Minutes	ست		أربعين	
	sit	Six	Ârbaçiyn	Forty
Hour Figures	٩		9	
	9	9	9	9
Minute Figures	٤٥		45	
	45	45	45	45

Table 3.11: Examples of entries for each Time gazetteer

The gazetteers built for the **Price NE extractor** are listed in Table 3.12 with the description and size in words for each gazetteer. The gazetteers in bold font were not originally built in NERA yet we have built them to facilitate the implementation of the rules and enhance the rules as well. Examples for the contents of each gazetteer mentioned in Table 3.12 are illustrated in Table 3.13 with transliteration and English translation.

Gazetteer name	Description	Size
Currency Name	List of currency names including the various spelling variations	453
Subcurrency	List of Subcurrency names including the various spelling variation	101
Power of Ten	List of words that represent the power of ten including the various spelling variations	41
Tens	List of numbers for tens in literal form including the various spelling variations	40
Units	List of numbers for units in literal form including the various spelling variations	75

Hundreds	List of numbers for hundreds in literal form including the various spelling variations	36
Thousands	List of numbers for thousands in literal form including the various spelling variations	31
Currency Location	List of locations, where the currency names belong to, including the various spelling variations	1275

Table 3.12: The gazetteers prepared for the Price extractor

Gazetteer name	**Examples of Entries**			
	Entry		**Entry**	
	Transliteration	**Translation**	**Transliteration**	**Translation**
Currency Name	درهم		يورو	
	dirham	Dirham	yuwruw	Euro
Subcurrency	فلس		القرش	
	fils	Fils	Alqirš	Penny
Power of Ten	مليون		مليار	
	milyuwn	One million	milyaAr	One billion
Tens	عشرون		تسعون	
	ςušruwn	Twenty	tisςuwn	Ninety
Units	أربعة		ثمانية	
	Ârbaçaħ	Four	ΘamaAnyaħ	Eight
Hundreds	تسعمائة		سبعمائة	
	tisςumaAŷaħ	Nine hundred	sabςumaAŷaħ	Seven hundred
Thousands	ألف		ألفين	
	Âalf	One thousand	Âalfayn	Two thousand
Currency Location	إماراتي		فرنسي	
	ĂmaAraAtiy	Emirati	faransiy	French

Table 3.13: Examples of entries for each Price gazetteer

The gazetteers built for the **Measurement NE extractor** are listed in Table 3.14 with the description and size in words for each gazetteer. The gazetteers in bold font were not originally built in NERA yet we have built them to facilitate the implementation of the rules and enhance the rules as well. Examples for the contents of each gazetteer mentioned in Table 3.14 are illustrated in Table 3.15 with transliteration and English translation.

Gazetteer name	Description	Size
Units1	List of known measurement units including the various spelling variations, e.g. متر (meter)	834
Units2	List of measures that appear at the beginning of units including the various spelling variation, e.g. "كيلو" (Kilo)	24
Units3	List of measurement units that is composed of two or more words/units separated by a backslash including the various spelling variations, e.g. "متر/ثانية" (meter/sec.)	70
Unit Suffix	List of words that may appear after measurement units including the various spelling variations, e.g. مكعب (cube)	59
Amount	List of numbers in literal form that describe amounts including the various spelling variations	767
Fractions	List of fractions in literal form including the various spelling variations	18
Power of Ten	List of words that represent the power of ten including the various spelling variations	85

Table 3.14: The gazetteers prepared for the Measurement extractor

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Units1	كيلومتر		جالون	
	kiyluwmitr	Kilometer	jaAluwn	Gallon
Units2	جيجا		نانو	
	jiyjaA	Giga	naAnuw	Nano
Units3	متر/ثانية		أمبير/متر	
	mitr/ΘaAniyaħ	Meter/second	Âambiyr/mitr	Ampere/meter
Unit Suffix	مكعب		مربع	
	mukaç~ab	Cube	murab~aç	Square
Amount	خمسة آلاف		مائتي	
	xamsaħ ĀlaAf	Five thousand	maAŷtay	Two hundred
Fractions	ربع		نصف	
	rubç	Quarter	niSf	Half
Power of Ten	مليارات		مئات	
	milyaAraAt	Billions	miŷaAt	Hundreds

Table 3.15: Examples of entries for each Measurement gazetteer

NERA has not been developed to recognize named entities of type Percent within Arabic text. Therefore, a Percent NE extractor has been developed as a part of the rule-based system in this research. The gazetteers built for the **Percent NE extractor** are listed in Table 3.16 with the description and size in words for each gazetteer. Examples for the contents of each gazetteer mentioned in Table 3.16 are illustrated in Table 3.17 with transliteration and English translation.

Gazetteer name	Description	Size
Percent Suffix	List of words/symbols that may appear after the Percent value including the various spelling variations	11
Amount	List of numbers in literal form that describe amounts including the various spelling variations	767
Fractions	List of fractions in literal form including the various spelling variations	18

Table 3.16: The gazetteers prepared for the Percent extractor

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Percent Suffix	بالمئة		%	
	biAlmiŷaħ	Percent	%	%
Amount	خمسة آلاف		مائتي	
	xamsaħ ĀlaAf	Five thousand	maAŷatay	Two hundred
Fractions	ربع		نصف	
	rubȿ	Quarter	niSf	Half

Table 3.17: Examples of entries for each Percent gazetteer

3.2.3. Gazetteers for Phone Number, File Name and ISBN Extractors

The gazetteers built for the **Phone Number NE extractor** are listed in Table 3.18 with the description and size in words for each gazetteer. The gazetteers in bold font were not originally built in NERA yet we have built them to facilitate the implementation of the rules and enhance the rules as well. Examples for the contents of each gazetteer mentioned in Table 3.18 are illustrated in Table 3.19 with transliteration and English translation.

Gazetteer name	Description	Size
Phone indicators1	List of words that may appear before phone numbers including the various spelling variations	17
Phone indicators2	List of words that should precede phone numbers including the various spelling variations	50
Phone indicators3	List of words that may appear before phone numbers (may refer to places and facilities) including the various spelling variations	58
Phone indicators4	List of words that may appear before phone numbers as adjectives including the various spelling variations	10
Phone indicators5	List of words that may appear before phone numbers as recommendations including various spelling variations	16
Relatives	List of Person relatives including the various spelling variations	121
Country	List of locations, where the phone number belong to, including the various spelling variations	1259

Table 3.18: The gazetteers prepared for the Phone Number extractor

Gazetteer name	Examples of Entries			
	Entry		Entry	
	Transliteration	Translation	Transliteration	Translation
Phone indicators1	هاتف		الرقم	
	haAtif	Telephone	Alraqam	The number
Phone indicators2	تليفون		موبايل	
	tiliyfuwn	Telephone	muwbaAyl	Mobile
Phone indicators3	المنزل		المكتب	
	Almanzil	The house	Almaktab	The office
Phone indicators4	الموحد		المباشر	
	AlmuwaH~ad	United	AlmubaAšir	Direct
Phone indicators5	للاستعلام		للاستفسار	
	lilAistiqlaAm	To inquire	lilAistifsaAr	To inquire
Relatives	الوالدة		أخيها	
	AlwaAlidaħ	The mother	ÂxiyhaA	Her brother
Country	قطري		سويسري	
	qatariy	Qatari	swiysriy	Swiss

Table 3.19: Examples of entries for each Phone number gazetteer

The File name extractor in NERA is originally an English File name extractor with some additions including Indic digits and Arabic characters. Thus, no details regarding the rules within the extractor are mentioned in the technical reports. Therefore, an Arabic File name extractor has been developed as a part of the rule-based system in this research. The gazetteers built for the **File name NE extractor** are listed in Table 3.20 with the description, examples and size in words for each gazetteer.

Gazetteer name	Description	Example	Size
Lowercase Ext.	List of file extensions with lowercase letters	zip	1205
Mix Cases Ext.	List of file extensions with mix of lowercase and uppercase letters	CATProcess	1183
Uppercase Ext.	List of file extensions with uppercase letters	DAT	1191

Table 3.20: The gazetteers prepared for the File name extractor

The ISBN extractor in NERA is originally an English ISBN extractor with some additions including Indic digits and Arabic characters. Thus, no details regarding the rules within the extractor are mentioned in the technical reports. Therefore, an Arabic ISBN extractor has been developed as a part of the rule-based system in this research. The gazetteer built for the **ISBN NE extractor** is:

ISBN Prefixes List of ISBN prefixes including ISBN and ردمك (ISBN).

3.3. Data Verification and Correction

During the tagging phase, errors of different kinds have been discovered within the original datasets. These errors can be classified into two categories including classification errors and spelling errors. The classification errors are mistakes in the classification or the tagging of an entity within the dataset, while spelling errors are mistakes in the spelling of

words. Table 3.21 demonstrates examples of the errors found in the corpora. The two kinds of errors may affect the results of the NER system and affect the system's capability in recognizing named entities with such errors.

Dataset	Error		Error category	Correction
	In Arabic	Translation		
ANERcorp	الموهوب O **تامر O** حبيب I-PERS	Talented O **Tamer O** Habeb I-PERS	Classification error	الموهوب O **تامر B-PERS** حبيب I-PERS
ANERcorp	نادي O **مانشستر I-ORG** يونايتد I-ORG	Club O **Manchester I-ORG** United I-ORG	Classification error	نادي O **مانشستر B-ORG** يونايتد I-ORG
ANERcorp	المستوى O الدولي O **فإن B-PERS** المجلس O	Level O International O **The B-PERS** Council O	Classification error	المستوى O الدولي O **فإن O** المجلس O
ACE 2004 BN	في ابريل **نسيان** الماضي	In last April **Nisan**	Spelling error	في ابريل **نيسان** الماضي
ACE 2003 NW	و كانت الأمم **المتحةد**	and **united** nations was	Spelling error	و كانت الأمم **المتحدة**
ACE 2004 NW	إف.بي.آي classified as Person name	FBI	Classification error	إف.بي.آي classified as Organization name
ACE 2003 BN	الأمين العام **للأم** المتحدة	Secretary General of the United **Nations**	Spelling error	الأمين العام **للأمم** المتحدة

Table 3.21: Examples of errors in the datasets

3.4. Data Preparation

Each corpus has gone through effective preparation steps before applying NER. The following subsections describe the tagging mechanism and the transformation of the corpora. However, a classification of the errors found in the reference datasets is illustrated. In this research, the tagging mechanism of the corpora varies from manual

tagging to automated tagging. The final produced datasets files are annotated and in XML format.

The **ACE corpora**, as stated earlier, have the annotations of the data in separate files. In our system, the input data file for training and testing should contain the annotations within the same file. Therefore, we have to combine the ACE data files with the corresponding Entity Information files so that one file is produced to represent each ACE dataset (i.e. an annotated data file). The tagging process of the ACE corpora was accomplished manually. Table 3.22 illustrates the categories of the tags contained in each produced ACE dataset file. It worth noting that the tagging of Person, Location and Organization named entities was based on the Entity Information files, while the tagging of other types of named entities (i.e. Temporal and Numerical expressions) was not necessarily based on Entity Information files since ACE 2003 and ACE 2004 do not have annotations for these types of named entities and ACE 2005 considers certain forms of temporal and numerical expressions and overlooks other forms. Figure 3.6 illustrates a sample of transformed ACE dataset in which our tag schema is used. Our tag schema, as mentioned in Chapter2, includes NE tags for person, location, organization, date, time, price, measurement, percent, phone number, ISBN and file name. The following are the 11 tags which a named entity will be enclosed with one of them:

<Person>*Entity*</Person>
<Location> *Entity*</Location>
<Organization> *Entity*</Organization>
<Date>*Entity* </Date>
<Time>*Entity* </Time>
<Price>*Entity*</Price>
<Measurement>*Entity*</Measurement>
<Percent>*Entity*</Percent>
<PhoneNumber>*Entity*</PhoneNumber>
<ISBN>*Entity*</ISBN>
<FileName>*Entity*</FileName>

Dataset		Categories of NE tags
ACE 2003	BN	Person, Location, Organization, Date, Time, Price, Measurement and Percent
	NW	Person, Location, Organization, Date, Time, Price, Measurement and Percent
ACE 2004	BN	Date, Time, Price, Measurement and Percent
	NW	Person, Location, Organization, Date, Time, Price, Measurement and Percent
ACE 2005	BN	Date, Time, Price, Measurement and Percent
	NW	Date, Time, Price, Measurement and Percent

Table 3.22: The distribution of the tag set over the ACE corpora

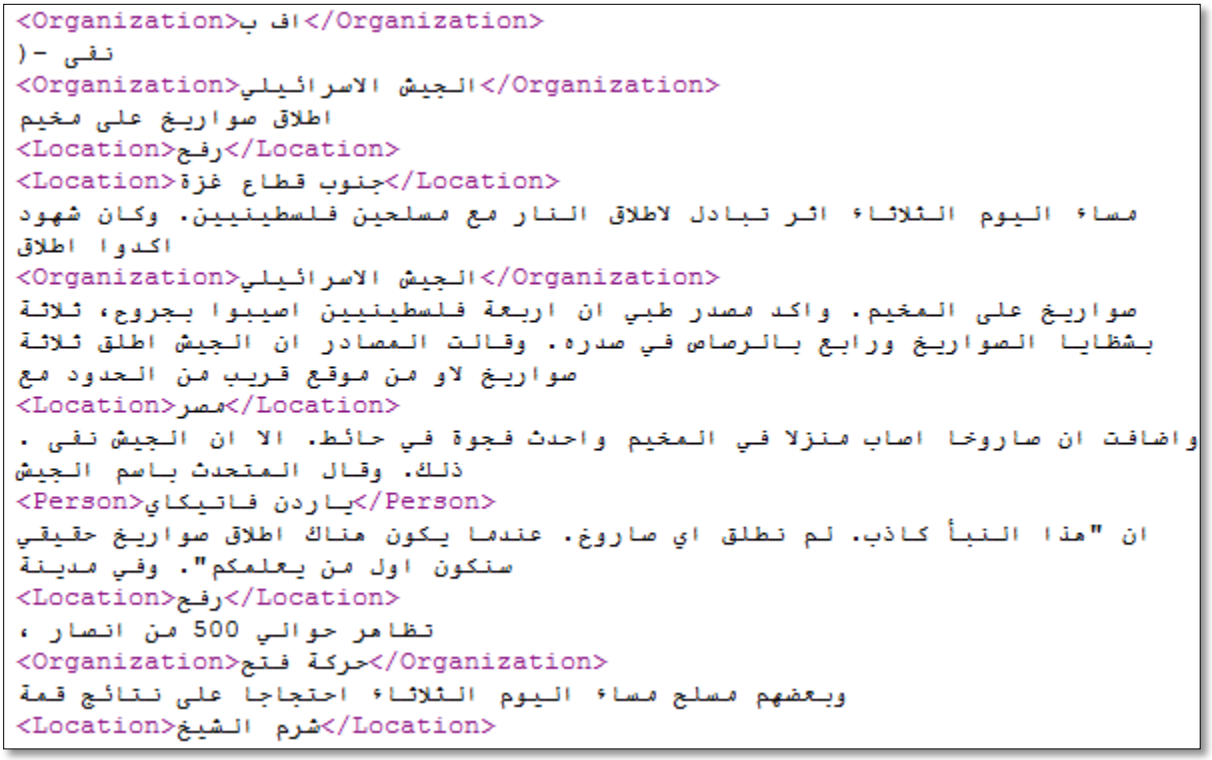

Figure 3.6: Sample of ACE 2003 NW transformed dataset

On the other hand, **ANERcorp dataset** has the annotations on the same file with the data in which the CoNLL tag set and IOB scheme are followed. We have developed a Java program which converts the schema of annotations from IOB schema to our tag schema; hence the tagging process of ANERcorp is considered as automated tagging mechanism. Only the Person, Location and Organization annotations in ANERcorp are considered in the transformation process, while Miscellaneous annotations are not of interest in this research. Figure 3.7 illustrates a sample of the ANERcorp dataset after the transformation.

Figure 3.7: Sample of ANERcorp transformed dataset

ATB part1 v 2.0 dataset and **our own corpus** have been annotated fully manually by the author of this thesis. The types of entities considered in the tagging process of ATB dataset are Date, Time, Price, Measurement and Percent, while Phone Number, ISBN and File Name are the types considered in the tagging process of our own corpus.

Table 3.23 illustrates the number of annotated named entities of different categories in each of the datasets used in this research.

NE type / Dataset		Person	Location	Organization	Date	Time	Price	Measurement	Percent	Phone Number	File Name	ISBN
ACE 2003	BN	711	1292	493	58	15	17	28	35			
	NW	517	1073	181	20	1	3	14	3			
ACE 2004	BN	1865	3449	1313	357	28	105	51	54			
	NW				67	4	36	30	32			
ACE 2005	BN				154	20	163	60	42			
	NW				37	7	9	22	5			
ANERcorp		3602	4425	2025								
ATB Part1 v 2.0					431	80	168	330	75			
Our own corpus										136	160	126
Total		**6695**	**10239**	**4012**	**1124**	**155**	**501**	**535**	**246**	**136**	**160**	**126**

Table 3.23: Number of different NEs annotated for NER in each dataset

3.5. Conclusion

In this research, linguistic resources of two types have been collected and analyzed carefully including gazetteers and datasets. The collected gazetteers are related in the first place with the grammatical rules used in the rule-based component. The proposed system utilizes various gazetteer sets including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name gazetteer sets. The methodology used for dataset collection is composed of five main phases. Firstly, acquiring and analyzing commercial datasets available for Arabic NER; secondly, analyzing freely available datasets for Arabic NER; thirdly, acquiring and developing other needed resources; fourthly, data verification and correction; finally, data preparation before applying NER. The datasets collected for this research are ACE corpora (i.e. ACE 2003, 2004 and 2005), ANERcorp, ATB Part1 v 2.0 and also our own corpus.

Chapter 4

The Architecture of the System - The Rule-Based Component

This chapter describes the architecture of the proposed hybrid NER system. The rule-based component is demonstrated and discussed in details. The implementation of the rules and the gazetteers within the GATE framework is explained in details.

4.1. The Proposed Hybrid System

In this research, A Hybrid NER system is proposed to tackle the problem of NER for Arabic. Figure 4.1 illustrates the architecture of our hybrid NER system for Arabic. The system consists of two main components including rule-based system and ML-based system. The hybrid system works on three main phases. Firstly, the rule-based NER phase; secondly, the feature selection and extraction phase; and finally, the ML-based NER phase.

The first phase involves running the rule-based NER system over an input text and then having as an output an annotated version of the input text. The main contribution of the rule-based component to the hybrid system is the generated annotations in which they are represented as a group of features called rule-based (RB) features. Additional types and groups of features are extracted in the second phase of the hybrid system such as morphological features and gazetteers features. The output of the second phase is a file with a record of features for each word in input text. The features file goes as an input to the ML-based system for training and testing purposes in case the actual NE classes are known. On the other hand, if the actual NE classes are unknown then the purpose is to predict the class of each word in the input text using a model (i.e. NER classifier) which has been learnt as a result of the training stage.

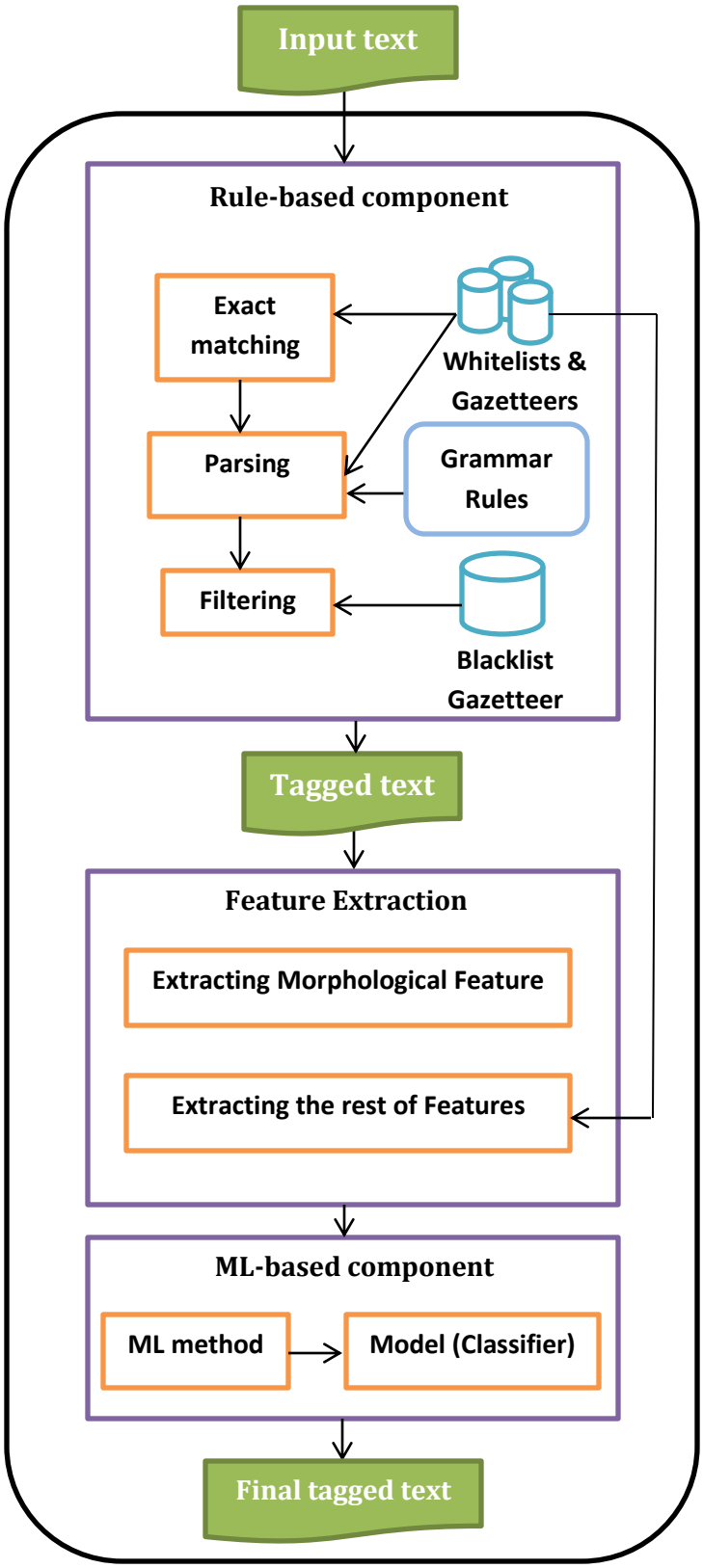

Figure 4.1: The Architecture of the Hybrid NER System

4.2. The Implementation of the NE Extractors

The rule-based component in our hybrid system is reproduction of the NERA system (Shaalan and Raza, 2008). The system is implemented in the GATE[1] framework. GATE facilitates the development of rule-based NER systems through providing the user with the capability of implementing grammatical rules as finite state transducer using JAPE. The main components of the GATE are the CREOLE, ANNIE and JAPE as described in section 2.5.1. The rule-based component is built with the capability of recognizing 11 different types of named entities including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name. The implementation of rules is based on the technical reports of NERA system except for Percent, ISBN and File Name rules. NERA has not been developed to recognize named entities of type Percent within Arabic text. Therefore in this research, a Percent NE extractor is developed as a part of the rule-based component. The File name extractor and the ISBN extractor in NERA are originally English extractors with some additions including Indic digits and Arabic characters. Thus, no details regarding the rules within the File name and ISBN extractors are mentioned in the technical reports. Therefore, an Arabic File name extractor and an Arabic ISBN extractor have been developed as parts of the rule-based system in this research.

The NERA system consists of three main components including Whitelists (i.e. lists of Full names of different types of NEs), Grammar Rules (i.e. in the form of regular expressions) and Filtration mechanism (i.e. blacklists of invalid NEs that have been detected via certain grammatical rules). A set of gazetteers/dictionaries are utilized as well within the NERA system.

4.2.1. Processing resources used in GATE

The rule-based component is built in the GATE environment as a corpus pipeline in which a corpus with documents is processed using different processing resources within GATE. The utilized processing resources are an Arabic Tokenizer, Gazetteers and Grammatical Rules as described below.

[1] Available for free download on http://gate.ac.uk/

The **Arabic tokenizer** is a built-in tool within GATE framework used to extract all tokens (i.e. words) from a document and indicates the type of the token such as string, punctuation, number, etc. The **Gazetteers** have been implemented as ANNIE gazetteers that are lists of named entities or keywords (i.e. triggers words) in which each line represents an entry within a gazetteer (all the gazetteers built in our system are described in Chapter 3, and ANNIE is described in section 2.5.1 of Chapter 2). The **Grammatical Rules** in GATE are built using JAPE Transducers that are finite state transducers based on regular expressions. JAPE[2] (i.e. Java Annotation Pattern Engine) is a pattern matching language produced to express grammatical rules as transducers. Each rule in a transducer consists of two sides including:

- **Lift Hand Side** (LHS): is the pattern specification of a rule in which annotations are assigned to the cases where the pattern is matched.
- **Right Hand Side** (RHS): is the label (i.e. the annotation) assigned to the matched expressions by the lift hand side specifications of a rule.

The implementation of all grammatical rules in this system is done through JAPE language in order to enable the recognition of the eleven types of named entities mentioned earlier.

4.2.2. Rule-based Named Entity Extractors

A description of each named entity extractor is given in the following subsections.

4.2.2.1. Person NE Extractor

The rules of the Person NE extractor has been implemented in JAPE by a related work (Abdallah, Shaalan and Shoaib, 2012) based on the technical report of NERA system. An example of a rule implementation within the Person NE extractor is given below.

Person Rule#5 in the form of regular expression:

((ابو|ام) + First Name + (First Name | Middle Name | Last Name)?)

[2] JAPE online documentation is available on http://gate.ac.uk/sale/tao/splitch8.html#x12-2050008

Description:

If the expression starts with "ابو" or "ام" and then followed by a First Person Name (i.e. belong to the First Names gazetteer) with the possibility of having a First Name, Middle Name or Last Name afterwards, the expression is identified as Person named entity.

Examples of Person names matched by Rule#5:

- ابو محمد (The father of Mohammed)
- ام خالد سعيد (The mother of Khaled Saeed)

Person Rule#5 in JAPE[3] language (as implemented in GATE):

```
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"}
```

It worth noting that the LHS of a JAPE rule is the part before "-->" sign, while the RHS is the part after "-->" sign.

4.2.2.2. Location NE Extractor

The rules of the Location NE extractor has been implemented in JAPE by a related work (Abdallah, Shaalan and Shoaib, 2012) based on the technical report of NERA system. An example of a rule implementation within the Location NE extractor is given below.

[3] GATE JavaDocs is available on http://jenkins.gate.ac.uk/job/GATE-Nightly/javadoc/index.html

Location Rule#6 in the form of regular expression:

((بـ)? + Administrative division + (State Name |Country Name |Any Location))

Description:

If the expression starts with an Administrative division as an indicator word, which may begin with "بـ", then followed by a State Name, Country Name or Any Location, the expression with the exclusion of the indicator word is identified as Location named entity.

Examples of Location names matched by Rule#6:

- مملكة البحرين (Kingdom of Bahrain)
- امبراطورية اليابان (Empire of Japan)

Location Rule#6 in JAPE language (as implemented in GATE):

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

4.2.2.3. Organization NE Extractor

The rules of the Organization NE extractor has been implemented in JAPE by a related work (Abdallah, Shaalan and Shoaib, 2012) based on the technical report of NERA system. An example of a rule implementation within the Organization NE extractor is given below.

Organization Rule#8 in the form of regular expression:

(Company preceding known part + '(' + unknown word +')')

Description:

If the expression starts with an organization prefix, which forms a part of the organization name, and followed by an unknown word within brackets then the expression is recognized as an Organization named entity.

Examples of Organization names matched by Rule#8:

- شركة سمارت فويل سل (Smar Foil Sell company)

Organization Rule#8 in JAPE language (as implemented in GATE):

```
Rule: Organization8
Priority: 30
(
{Lookup.majorType=="company_preceding_known_part"}
{Token.string=="("}
{Token}
{Token.string==")"}
):Org
-->
:Org.Organization = {rule="OrganizationRule8"}, :Org.organization =
{rule="OrganizationRule8"}
```

4.2.2.4. Date NE Extractor

In this research, the rules of the Date NE extractor have been implemented in JAPE language upon the technical reports of NERA system as part of the rule-based component. Some of the rules have been modified to improve the performance and the accuracy of the

Date NE extractor based on the contextual analysis. An example of a rule implementation within the Date NE extractor is given below.

Date Rule#1 in the form of regular expression (the original version):

((Weekday)? + (DayFigArabic-Indic|DayFigArabic) + ((و + (DayFigEng|DayFigArabic))| من)? + (MonthName ([-/] (MonthName))?) + ([-/] (من|في)? (سنة|العام)? + (yearFigArabic|yearFigArabic-Indic|relativeDateWord|yearWord) (year_range)?))

Date Rule#1 in the form of regular expression (the modified version):

((Weekday)? + (DayFigArabic-Indic|DayFigArabic) + ((و|الى|إلى)+(DayFigEng|DayFigArabic))? (من)? + (MonthName ([-/] (MonthName))?) + ([-/] (من|في)? (عام|السنة|سنة|العام)? + (yearFigArabic|yearFigArabic-Indic|relativeDateWord|yearWord)? (year_range)?) (year_type)?)

Description:

This rule identifies Date named entities in the format "*day month year*" which may start with a weekday name. The days are represented with digital figures, while the months with words. The years may be in digital form or literal form. The day can be followed by "و" (and), "إلى" or "الى" (to) then another day. An optional "من" (from) preceding a month which may be followed by another month and an optional date separator (- or /) between the two months. Then there is a year which is optionally preceded by a date separator (- or /) or a specific word such as ("من" or "في") (from or in) followed by the word ("العام", "عام", "سنة" or "السنة") (year). The year may be followed by a year range and a year type.

The Modifications (as highlighted in red color in the modified version of regular expression):

- Adding the word "إلى" and it spelling variant "الى" so that they may precede a second day. E.g. 8 إلى 10 يناير (8 to 10 January).
- Making the first "من" (from) separated and optional so that the date expression "*day* and *day* from *month*" can be detected. E.g. 1 و 2 من مايو (1 and 2 from May).
- Adding "السنة|عام" to (سنة|العام) so that spelling variation is considered by the rule.

– Adding the year type to the end of the rule to enable the recognition of dates that end with a year type. E.g. 7 مـايو مـن العـام 2010 ميلاديـة (7 May of the Gregorian year 2010).

Examples of dates matched by Rule#1:

– السبت 8 كانون الثاني من العام 2000 (Saturday 8 Kanoun the second of the year 2000)
– 22 مايو الماضي (22 of last May)
– 17 و 18 يوليو (17 and18 June)
– مـايو مـن سنة ١٩٩٢ وحتـى نهايـة عـام ٢٠٠٠ (May from the year 1992 till the end of year 2000)

Date Rule#1 in JAPE language (as implemented in GATE):

```
Rule: DateRule1
Priority: 10
(  ({Lookup.majorType == "Ar_weekday_d"})?
 {Lookup.majorType == "Ar_dayFig_d"}
 ( ( ({Token.string == "و"}|{Token.string == "إلى"}|{Token.string == "الى"})
    ({Lookup.majorType == "Ar_dayFig_d"})|{Lookup.majorType == "Ar_andDayFig_d"})  )?
  ({Token.string == "من"})?
 {Lookup.majorType == "Ar_month_d"}
  ( ({Token.string == "-"}|{Token.string == "/"}|{Token.string == "("})?
   {Lookup.majorType == "Ar_month_d"}({Token.string == ")"})? )?
 ({Token.string == "-"}|{Token.string == "/"})? ((({Token.string == "في"}|{Token.string == "من"})
  ({Token.string == "عام"}|{Token.string == "العام"}|{Token.string == "سنة"}|{Token.string == "السنة"}))|
 ({Token.string == "عام"}|{Token.string == "العام"}|{Token.string == "سنة"}|{Token.string == "السنة"}))?
 ( {Token.kind == "number"}|{Lookup.majorType == "Ar_relativeWord_d"}|(YEAR_WORD) )?
 ({Lookup.majorType == "Ar_yearType_d"})?  ( {Lookup.majorType == "Ar_yearRange_d"}
  ({Token.string == "عام"}|{Token.string == "العام"}|{Token.string == "سنة"}|{Token.string == "السنة"})?
  ( {Token.kind == "number"}|{Lookup.majorType == "Ar_relativeWord_d"}|(YEAR_WORD) )
  ( {Lookup.majorType == "Ar_yearType_d"} )? )?  ):Dat
-->
:Dat.Date = {rule = "DateRule1"}
```

It worth noting that a year in the form of words is detected through a Macro embedded within the Date extractor to enable detecting Arabic year in literal form. The Macro is called "YEAR_WORD".

4.2.2.5. Time NE Extractor

In this research, the rules of the Time NE extractor have been implemented in JAPE language upon the technical reports of NERA system as part of the rule-based component. Some of the rules have been added or modified to improve the performance and the accuracy of the Time NE extractor. An example of a rule implementation within the Time NE extractor is given below.

Time Rule#6 in the form of regular expression (the original version):

(الساعة)? + (HourFig|HourInWord) + (الا|إلا) + (MinFig|MinInWord) + دقائق

Time Rule#6 in the form of regular expression (the modified version):

(الساعة|الساعه)? + (HourFig|HourInWord) + (و|الا|إلا) + (MinFig|MinInWord) + (دقائق|دقيقة|دقيقه)

Description:

This rule identifies Time in the format of hour either in literal form or digital form followed by the word ("و" (and), "الا" or "إلا" (to)) and then minutes either in literal form or digital form. The expression may start with the word ("الساعه" or "الساعة") (o'clock).

The Modifications (as highlighted in red color in the modified version of regular expression):

- Adding the word "الساعه" as another spelling variation of the word "الساعة" (o'clock).
- Adding the conjunction "و" (and) to the word group ("إلا" or "الا") (to) so that the rule can recognize the time expression "*Hour* and *minutes*". E.g. الساعة الخامسة و أربع دقائق (Five o'clock and four minutes)

– Adding optional words (“دقيقـة” and “دقيقـه”) (minute) to “دقـائق” (minutes) as (“دقـائق”, “دقيقـة” or “دقيقـه”) (minutes or minute) to enable the recognition time in certain format such as الساعة الخامسة و عشرون دقيقة (Five o’clock and twenty minutes).

Examples of Time matched by Rule#6:

– الساعة 7 و 15 دقيقه (7 o’clock and 15 minutes)
– 8 إلا 5 دقائق (5 minutes to 8)

Time Rule#6 in JAPE language (as implemented in GATE):

```
Rule: TimeRule6
Priority: 15
(
 ({Token.string == "الساعة"}|{Token.string == "الساعه"})
 (({Lookup.majorType == "Ar_hourInWord_t"} ({Lookup.majorType == "Ar_hour_tens_t"})? )
  |{Lookup.majorType == "Ar_hourFig_t"})
 ({Token.string == "إلا"}|{Token.string == "الا"}|{Token.string == "و"})?
 (({Lookup.majorType == "Ar_minInWord_t"} ({Lookup.majorType == "Ar_tens_t"})? )
  |{Lookup.majorType == "Ar_minFig_t"})
 ({Token.string == "دقائق"}|{Token.string == "دقيقة"}|{Token.string == "دقيقه"})
 )
:Tim
-->
:Tim.Time = {rule = "TimeRule6"}
```

4.2.2.6. Price NE Extractor

In this research, the rules of the Price NE extractor have been implemented in JAPE language upon the technical reports of NERA system as part of the rule-based component. Some of the rules have been modified to improve the performance and the accuracy of the Price NE extractor. An example of a rule implementation within the Price NE extractor is given below.

Price Rule#4 in the form of regular expression (the original version):

(Power of Ten + Currency Name)

Price Rule#4 in the form of regular expression (the modified version):

(Power of Ten + (Currency Name | Subcurrency Name))

Description:

This rule matches a price expression which consists of an amount represented by power of ten followed by a Currency name or a Subcurrency name.

The Modifications (as highlighted in red color in the modified version of regular expression):

- Adding the possibility of having an amount represented by power of ten and followed by a Subcurrency name instead of a Currency name. E.g. 100 قرش (100 penny)

Examples of Price matched by Rule#4:

- ألفا درهم (Two thousand Dirhams)
- مئة سنت (One hundred cent)

Price Rule#4 in JAPE language (as implemented in GATE):

```
Rule: PriceRule4
Priority:43
( {Lookup.majorType == "Ar_power_of_ten_p"}
  ({Lookup.majorType == "Ar_currency_name_p"}|{Lookup.majorType == "Ar_subcurrency_p"}) )
:Prce
-->
:Prce.Price = {rule = "PriceRule4"}
```

4.2.2.7. Measurement NE Extractor

In this research, the rules of the Measurement NE extractor have been implemented in JAPE language upon the technical reports of NERA system as part of the rule-based component. Some of the rules have been modified to improve the performance and the accuracy of the Measurement NE extractor. An example of a rule implementation within the Measurement NE extractor is given below.

Measurement Rule#2 in the form of regular expression (the original version):

(NumFig|NumWord) + [xX] + (NumFig|NumWord)

Measurement Rule#2 in the form of regular expression (the modified version):

(NumFig|NumWord)+ (Fraction)? + [xX×] + (NumFig|NumWord) + (Fraction)?

Description:

This rule matches a Measurement expression which consists of an amount in literal form or digital form followed by a multiplication sign (x, X or ×) then another amount in literal form or digital form in the end. The amounts may be followed by fractions.

The Modifications (as highlighted in red color in the modified version of regular expression):

- Adding the possibility of having fractions after the numerical amounts.
- Considering another variation of the multiplication sign which is "×".

Examples of Measurement matched by Rule#2:

- 1024x768
- واحد ونصف × ثلاثة (one and a half × three)

Measurement Rule#2 in JAPE language (as implemented in GATE):

```
Rule: MeasurementRule2
Priority: 11
(
 ({Token.kind == "number"}|(AMOUNT_WORDS))
 ( {Lookup.majorType == "Ar_fraction_m"} (AMOUNT_WORDS)? )?
 ({Token.string == "x"}|{Token.string == "X"}|{Token.string == "×"})
 ({Token.kind == "number"}|(AMOUNT_WORDS))
 ( {Lookup.majorType == "Ar_fraction_m"} (AMOUNT_WORDS)? )?
)
:Measure
-->
:Measure.Measurement = {rule = "MeasurementRule2"}
```

It worth noting that an amount in the form of words is detected through a Macro embedded within the Measurement extractor to enable detecting Arabic numerical amounts in literal form. The Macro is called “AMOUNT_WORDS”.

4.2.2.8. Percent NE Extractor

In this research, the rules of the Percent NE extractor have been implemented in JAPE language upon our analysis of Arabic text which has led to extract rules for matching Percent expressions as part of the rule-based component. An example of a rule implementation within the Percent NE extractor is given below.

Percent Rule#1 in the form of regular expression:

```
( (NumFig|NumWord) + ( (الا|إلا|و)? + Fraction + (Power of Ten)? )? +
((و|إلى|الى|او|أو) (NumFig|NumWord) )? + Percent Suffix )
```

Description:

This rule matches Percent expressions that start with an amount in literal form or digital form followed by an optional fraction preceded by the words (“إلا”, “الا” (to) or “و” (and)), then followed by an optional amount either in literal form or digital form preceded by the words (“أو”, “او” (or), “إلـى”, “الـى” (to) or “و” (and)). The expression ends with a Percent Suffix.

Examples of Percent expression matched by Rule#1:

- 99%
- خمسون بالمئة (Fifty percent)

Percent Rule#1 in JAPE language (as implemented in GATE):

```
Rule: PercentRule1
Priority:10
(
  (({Token.kind == "number"} (({Token.string == "."}|{Token.string == "ر"}|{Token.string ==
","}|{Token.string == "،"})      {Token.kind == "number"})? )|(AMOUNT_WORDS))
 ( ({Token.string == "الا"}|{Token.string == "إلا"}|{Token.string == "و"})
  {Lookup.majorType == "Ar_fraction_m"} ({Lookup.majorType == "Ar_power_of_ten_m"})?
({Token.string == ")"})?)?
 ( ({Token.string == "أو"}|{Token.string == "او"}|{Token.string == "و"}|{Token.string ==
"إلى"}|{Token.string == "الى"})
  ({Token.string == "("})? (({Token.kind == "number"} (({Token.string == "."}|{Token.string ==
"ر"}|{Token.string == ","}
  |{Token.string == "،"})      {Token.kind == "number"})? )|(AMOUNT_WORDS)) ({Token.string ==
")"})? )?
( ({Token.string == "بالمئة"}|{Token.string == "بالمائة"}|{Token.string == "بالمائه"}|{Token.string ==
"بالمئه"}|{Token.string == "%"}) |( ({Token.string == "في"}|{Token.string == "فى"}) ({Token.string ==
"المئة"}|{Token.string == "المئه"}|{Token.string == "المائة"}|{Token.string == "المائه"}) )|( {Token.string ==
"نسبة"} ({Token.string == "مئوية"}|{Token.string == "مئويه"}) )  )      ):Perc
-->
:Perc.Percent = {rule = "PercentRule1"}
```

It worth noting that an amount in the form of words is detected through a Macro embedded within the Percent extractor to enable detecting Arabic numerical amounts in literal form. The Macro is called “AMOUNT_WORDS”.

4.2.2.9. Phone Number NE Extractor

In this research, the rules of the Phone Number NE extractor have been implemented in JAPE language upon the technical reports of NERA system as part of the rule-based component. Some of the rules have been added or modified to improve the performance and the accuracy of the Phone Number NE extractor. The following is an example of a new rule implemented within the Phone Number NE extractor based on our analysis.

Phone Number (New) Rule#8 in the form of regular expression:

((+) + 3 digits + (- | whitespace) + (<=3 digits) + (- | whitespace) + (>3 digits))

Description:

This rule matches Phone Numbers in the gulf region which consists of three groups of digits (Arabic or Hindi digits). The expression starts with “+” sign followed by 3 digits, then 3 digits or less, and more than 3 digits in the end. The digit groups are separated by “-” or a whitespace.

Examples of Phone Number matched by Rule#8:

- +971 50 6111234
- +971-55-2331965
- +973-78-9654

Phone Number Rule#8 in JAPE language (as implemented in GATE):

```
Rule: PhoneNumberRule8
Priority: 17
(
 ({SpaceToken.kind == "space"}|{Token.string == ":"})
 (  {Token.string == "+"}  {Token.kind == "number", Token.length == "3"}
    ({Token.string == "-"}|{SpaceToken.kind == "space"})
    {Token.kind == "number", Token.length <= "3"} ({Token.string == "-"}|
     {SpaceToken.kind == "space"})
    {Token.kind == "number", Token.length > "3"}   ):Phone8
)
:Phone_r
-->
:Phone8.PhoneNumber = {rule = "PhoneNumberRule8"}
```

4.2.2.10. File Name NE Extractor

In this research, the rules of the File Name NE extractor have been implemented in JAPE language upon the analysis of Arabic text and the File Name NE examples given by technical reports of NERA system (no rules are mentioned in the technical reports) as part of the rule-based component. An example of a rule implementation within the File Name NE extractor is given below.

File Name Rule#1 in the form of regular expression:

```
(   (( (Word|Digits) + (:|_|-|/|\)* )+) +  (( (Word|Digits) + (:|_|-|/|\)* )+) + “.” +
    (LowerCase_Ext|UpperCase_Ext|MixCases_Ext)   )
```

“” means zero or more than one time.*

“+” means one or more times.

Description:

This rule matches a File Name which consists of words/numbers with optional file name separators (:, -, _, / or \) to form the name of the file that is followed by a dot "." and an extension abbreviation. The extension may be in uppercase, lowercase or mixture of lower and upper cases.

Examples of File Name matched by Rule#1:

- My_picture.png
- 01-folder.zip

File Name Rule#1 in JAPE language (as implemented in GATE):

```
Rule: FileNameRule1
Priority:10
(
 (
  ( ((W_N) ({Token.string == ":"}|{Token.string == "_"}|{Token.string == "-"}|{Token.string == "/"}
        |{Token.string == "\\"})*  )+
    {Token.string == "."} )*
  ((W_N) ({Token.string == ":"}|{Token.string == "_"}|{Token.string == "-"}|{Token.string == "/"}
        |{Token.string == "\\"})*  )+
   {Token.string == "."}
  ({Lookup.majorType == "LowerCase_ext"}| {Lookup.majorType == "UpperCase_ext"}|
    {Lookup.majorType == "MixCases_ext"}) ):fn
   {!Token.string == "/", !Token.string == "\\", !Token.string == "."}
)
:fn_1
-->
:fn.FileName = {rule = "FileNameRule1"}
```

4.2.2.11. ISBN NE Extractor

In this research, the rules of the ISBN NE extractor have been implemented in JAPE language upon the analysis of Arabic text and the ISBN NE examples given by technical reports of NERA system (no rules are mentioned in the technical reports) as part of the rule-based component. An example of a rule implementation within the ISBN NE extractor is given below.

ISBN Rule#2 in the form of regular expression:

((ISBN|isbn|ردمك) + (":")? + (10 digits | 13 digits))

Description:

This rule matches an ISBN NE which consists of 10 digits or 13 digits preceded by ISBN prefix (ISBN, isbn or ردمك (ISBN)) followed by optional colon ":". The ISBN prefix is not part of ISBN NE; only the digits are labeled.

Examples of ISBN matched by Rule#2:

- ردمك: 9786038058558 (ISBN: 9786038058558)
- ISBN 9786030082438

ISBN Rule#2 in JAPE language (as implemented in GATE):

```
Rule: ISBNRule2
Priority:11
( ({Token.string == "isbn"}|{Token.string == "ISBN"}|{Token.string == "ردمك"}) ({Token.string == ":"})?
  (    (({Token.kind == "number", Token.length == "10"})|
       ({Token.kind == "number", Token.length == "13"}))    ):ISBN_e    )
:ISBN_r
-->
:ISBN_e.ISBN = {rule = "ISBNRule2"}
```

4.3. Integration of Different NE Extractors

After all the processing resources have been built, they are integrated into one application within GATE environment as illustrated in Figure 4.2. Table 4.1 illustrates the number of gazetteers and rules implemented within each NE extractor. In order to run the application over a dataset, a corpus should be created under language resources section in GATE. Then, the dataset can be imported into the created corpus with encoding "UTF-8" so that the Arabic script is correctly interpreted. The imported data file may be of different formats such as XML, HTML and RTF. In this research, the dataset are preprocessed and converted into XML format. The result of running the application (i.e. the annotations) can be observed through GATE interface as the example illustrated in Figure 4.3 or through saving the file as a XML file with annotations preserved. It worth noting that the annotation set is defined through the written rules. The annotation set used by our system is described in Chapter 2 and Chapter 3.

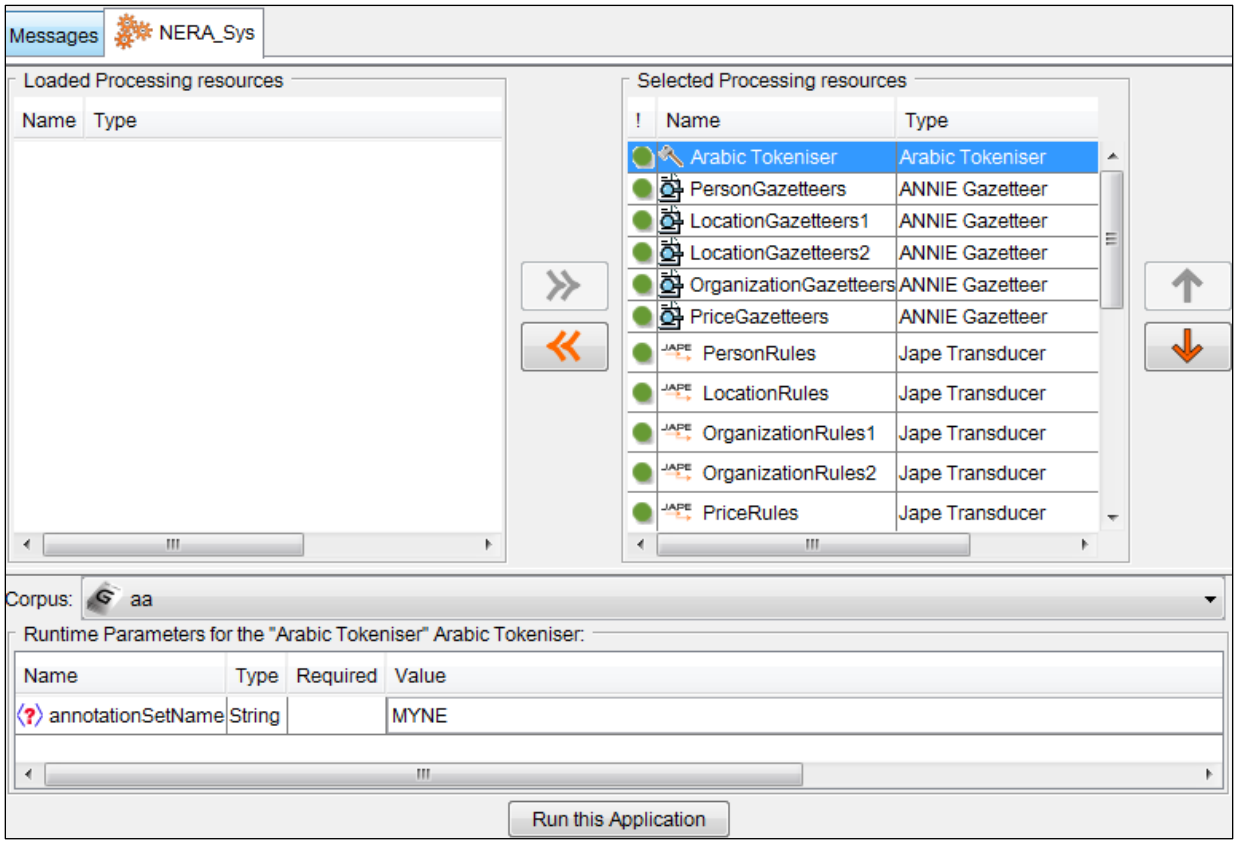

Figure 4.2: Grouping the processing resources under one application in GATE to form the rule-based system

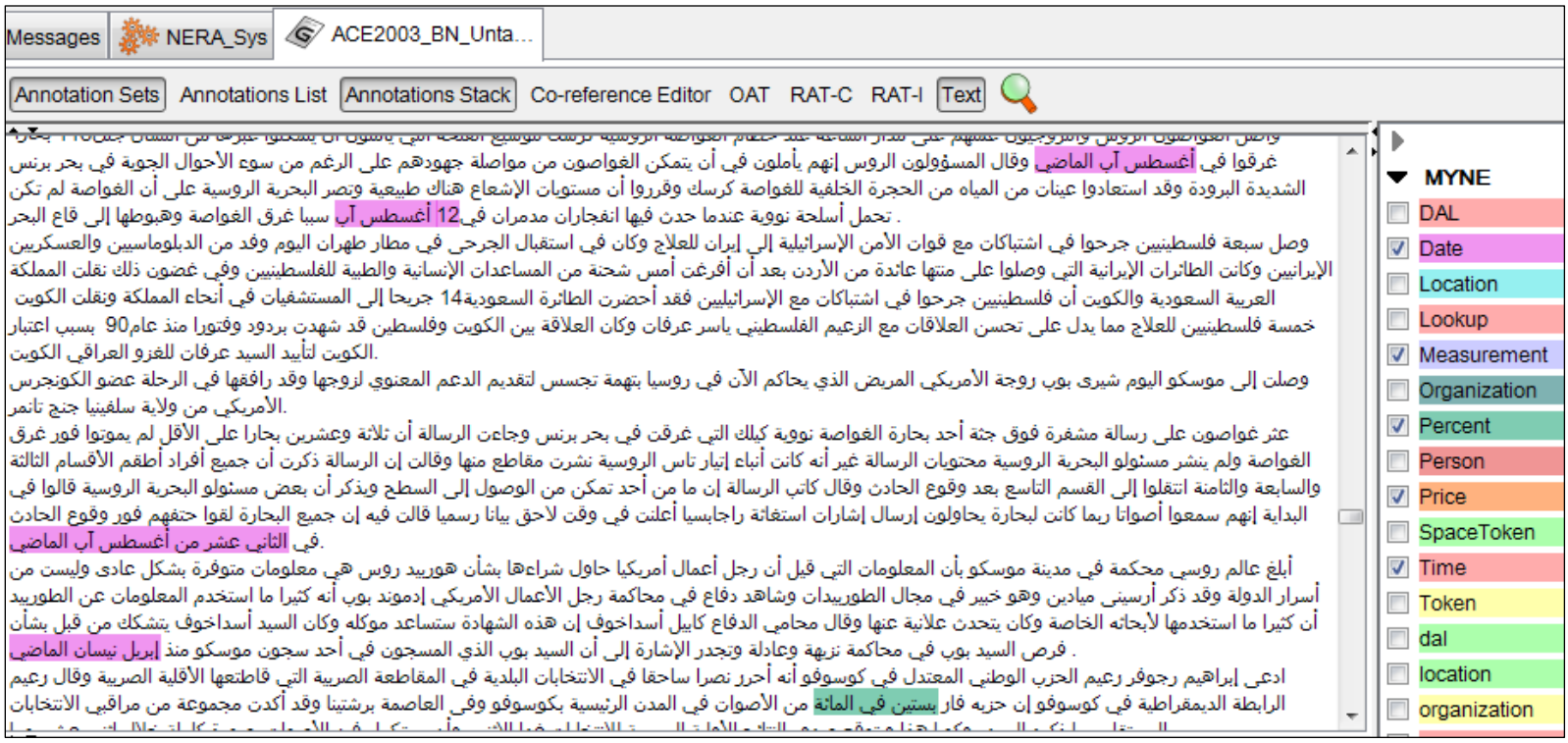

Figure 4.3: The annotations as appear in GATE interface when the system is applied on ACE 2003 BN dataset – Temporal and Numerical expressions are selected to appear on the GATE interface

	Person	Location	Organization	Date	Time	Price	Measurement	Percent	Phone Number	File Name	ISBN	Total
Number of Gazetteers	11	20	8	12	10	8	7	3	7	3	1	**90**
Number of Rules	9	20	9	7	8	4	3	1	7	3	2	**73**

Table 4.1: The Number of Gazetteers and Rules in each NE Extractor

4.4. Conclusion

In this research, the methodology followed to construct the rule-based component is composed of four main steps. Firstly, re-implementing a previous Arabic rule-based NER system (Shaalan and Raza, 2008) which was originally implemented within a commercial tool; instead, GATE, a freely available developmental tool, has been used to build the rule-based component. Secondly, verifying the grammatical rules. Thirdly, adjusting the grammatical rules with the necessary modifications in order to enhance the performance of the NE extractors. Finally, adding new grammatical rules that are derived from contextual analysis of Arabic texts. The implemented rule-based component is capable of recognizing NEs of different types including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name.

Chapter 5

The Architecture of the System - The ML-Based Component

This chapter describes the ML component is in details. The feature set and mechanism of extracting features are demonstrated with their architectures. The ML approaches utilized in the ML-based component are highlighted. The tools used in the feature extraction phase and the ML-based system are overviewed as well. The architecture of the ML-based component considering training and prediction (i.e. classification) phases is illustrated.

5.1. Machine Learning Approaches Tool – WEKA

WEKA[1], which stands for *Waikato Environment for Knowledge Analysis*, is a comprehensive workbench with a set of ML algorithms exploited for data mining. WEKA provides the environment to apply different ML techniques effectively on datasets in order to learn ML-based models. In this research, WEKA is utilized as the environment of ML-based component where different ML approaches are selected to be exploited in our hybrid NER system. The format of the produced feature set data files in the previous phase (i.e. feature extraction phase) is made compatible with WEKA workbench specifications.

5.2. The Selected Machine Learning Approaches

In this research, three different ML approaches have been utilized and examined individually as part of the ML-based component of our hybrid NER system including:

- **Decision trees approach** which is applied using J48 classifier in WEKA, an overview of decision trees is given in chapter 2 – subsection 2.6.2.1.

[1] The official website of WEKA is www.cs.waikato.ac.nz/ml/weka/

- **Support vector machines approach** which is applied using Libsvm classifier in WEKA, an overview of support vector machines is given in chapter 2 – subsection 2.6.2.1.

- **Logistic regression approach** which is applied using Logistic classifier in WEKA. Logistic regression is about modeling the probability of each class in a predefined set of classes (i.e. NE classes in this research) through linear functions. Logistic models are optimized through maximizing the conditional likelihood. More information regarding logistic regression algorithm can be found in Hastie, Tibshirani and Friedman (2009).

5.3. The Architecture of the ML-based Component

The output of the Feature Extraction Phase (section 5.4) is utilized as the input to the ML-based component where the feature set data files are used to learn classification models. The phase of learning a statistical model is called the training phase. The training phase in our system is illustrated in Figure 5.1. Recall, Figure 4.1 demonstrates the architecture of our hybrid system which illustrates the position of the ML-based component in the whole process.

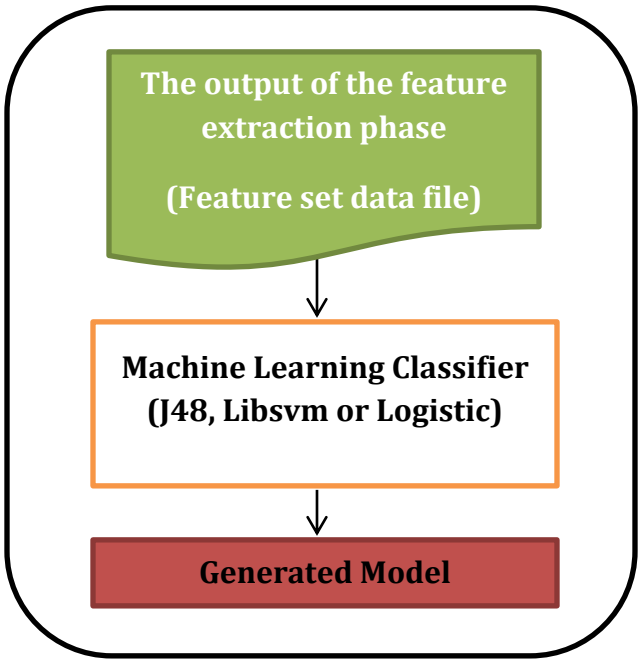

Figure 5.1: The Architecture of the Training Phase

After the training phase is completed, a classification model is generated by the classifier based on the features of the input dataset. The model is then used for the prediction task. The input in the prediction phase is mainly the features of a previously unseen data set in which the actual class of each word is unknown. The model generated by the training phase is utilized within the prediction phase to predict a class for each word in the input dataset. Figure 5.2 illustrates the architecture of the prediction phase.

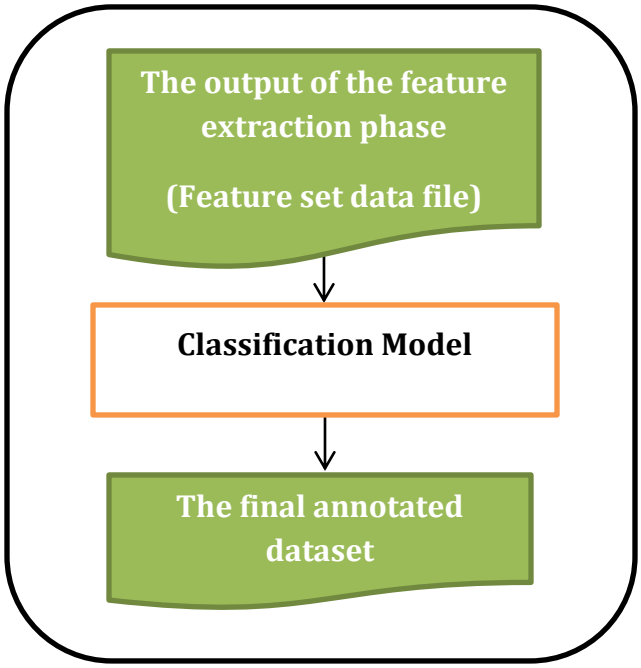

Figure 5.2: The Architecture of the Prediction Phase

5.4. Feature Set and Feature Extraction

In ML-based systems, suitable ML techniques and feature sets have to be selected in order to generate models with high performance and accuracy. The feature set is a very critical and important aspect in Machine Learning because ML-based models observe the data in an input through the features representing each element in the input data. More information regarding feature set can be found in Chapter 2 - subsection 2.6.2.2. In this research, the explored features can be divided into various types of features including rule-based features, morphological features, POS features, Gazetteer features, contextual features within the Gazetteers features, and other individual contextual and word-level features.

Exploring different types of features allows studying the effect of each feature type on the performance of the proposed hybrid NER system on different dimensions including named entity type and ML technique.

The category of **rule-based features** is the main contribution of the rule-based component to the hybrid system in which the annotations produced by rule-based component are utilized as features within the feature set. A window of words is a group of certain number of adjacent words centered by a targeted word. The annotations of a window of five words centered by the encountered word in an input text represent the rule-based features in each vector of features.

Due to the complex nature of Arabic morphology, investigating the **morphological features** can help observing patterns related to the morphology of words in Arabic to enable NER. The MADA[2] system is exploited as the morphological analyzer in our system. MADA generates 14 morphological features per input word. One of MADA features is POS tag which we have used to form another group of features called POS features. A named entity is usually a noun or proper noun; hence POS features are indicators for named entities. The other 13 features represent the morphological features in the feature set. More information regarding the MADA and the 14 features can be found in Chapter 2 – subsection 2.5.3. **POS features** are composed of the POS tag and features that relate POS tags to NER such as whether the POS tag is proper noun, noun or number words.

The gazetteers built as part of the rule-based system are also exploited within the phase of feature extraction to produce the **Gazetteer features**. Mainly these features represent the existence of a word within one or more of the relative groups of gazetteers which mau give an indication of a named entity. The gazetteers features of the immediate right neighbor and immediate left neighbor are considered as part of the feature set in general and the gazetteer features in particular for a candidate word. The features of neighbors are considered **contextual features** on the level of Gazetteers features.

The **other features** in the feature set are the word length (i.e. a word-level feature), the gloss of the word (i.e. a word-level feature) in which the English translation of a candidate is generated to check if the translation starts with capital letter as a possible indication of encountering a named entity, and the Dot presence on the left or the right of a word (i.e.

[2] MADA is Available for free download on http://www1.ccls.columbia.edu/MADA/MADA_download.html

contextual feature). It worth noting that the English gloss of each word in the input text is generated using MADA.

The 11 categories of named entities that our system can extract from Arabic scripts are distributed among three groups according to their nature in which each group has a distinct feature set:

- 1st group: Person, Location and Organization (which is known as ENAMEX).
- 2nd group: Date, Time, Price, Measurement and Percent (which is known as TIMEX and NUMEX)
- 3rd group: Phone Number, ISBN and File Name (the first two types of NE can be considered as NUMEX but they have been added to this group intentionally because of the nature of their rules and patterns which is specific and limited)

5.4.1. General Feature Set

The common set of features between the three groups are the following:

- Rule-based features: the NE class predicted by the rule-based module for the targeted word and the NE classes for the two immediate left neighbors and the two immediate right neighbors of the candidate word.
- POS tag: part-of-speech tag of the targeted word estimated by MADA.
- Morphological Features: set of 13 features generated by MADA.
- Check the word length flag: if the length is greater than or equals three then the value of this feature for this word is True and False otherwise.
- Check dot presence flag (i.e. preceding or following dot flag): if there is a dot '.' on the left or the right of the targeted word then the feature value is True and False otherwise.
- Check capitalized gloss flag: if the English translation of the targeted word begins with capital letter then the feature value is True and False otherwise.
- Actual NE tag of the targeted word.

5.4.2. Feature set of the 1st group

The feature set of the 1st group includes the following features:

- Check POS feature flags:
    - If POS is Noun then the value of this feature for this word is True and False otherwise.
    - If POS is Proper Noun then the value of this feature for this word is True and False otherwise.
- Check Gazetteers feature flags (a set of nine Boolean features):
    - Check Person Gazetteer:
        - If the targeted word belongs to Person Gazetteer then the feature value is True and False otherwise.
        - If the left neighbor of targeted word belongs to Person Gazetteer then the feature value is True and False otherwise.
        - If the right neighbor of targeted word belongs to Person Gazetteer then the feature value is True and False otherwise.
    - Check Location Gazetteer:
        - If the targeted word belongs to Location Gazetteer then the feature value is True and False otherwise.
        - If the left neighbor of targeted word belongs to Location Gazetteer then the feature value is True and False otherwise.
        - If the right neighbor of targeted word belongs to Location Gazetteer then the feature value is True and False otherwise.
    - Check Organization Gazetteer:
        - If the targeted word belongs to Organization Gazetteer then the feature value is True and False otherwise.

- If the left neighbor of targeted word belongs to Organization Gazetteer then the feature value is True and False otherwise.
- If the right neighbor of targeted word belongs to Organization Gazetteer then the feature value is True and False otherwise.

5.4.3. Feature set of the 2nd group

The feature set of the 2nd group includes the following features:

- Check POS feature flags:
  - If POS tag is Noun_num (i.e. number word) then the value of this feature for this word is True and False otherwise.
  - If POS tag is Proper Noun then the value of this feature for this word is True and False otherwise.
- Check Gazetteers feature flags (set of 15 Boolean features):
  - Check Date Gazetteers:
    - If the targeted word belongs to Date Gazetteers then the feature value is True and False otherwise.
    - If the left neighbor of targeted word belongs to Date Gazetteers then the feature value is True and False otherwise.
    - If the right neighbor of targeted word belongs to Date Gazetteers then the feature value is True and False otherwise.
  - Check Time Gazetteers:
    - If the targeted word belongs to Time Gazetteers then the feature value is True and False otherwise.
    - If the left neighbor of targeted word belongs to Time Gazetteers then the feature value is True and False otherwise.

- If the right neighbor of targeted word belongs to Time Gazetteers then the feature value is True and False otherwise.

- Check Price Gazetteers:
  - If the targeted word belongs to Price Gazetteers then the feature value is True and False otherwise.
  - If the left neighbor of targeted word belongs to Price Gazetteers then the feature value is True and False otherwise.
  - If the right neighbor of targeted word belongs to Price Gazetteers then the feature value is True and False otherwise.

- Check Measurement Gazetteers:
  - If the targeted word belongs to Measurement Gazetteers then the feature value is True and False otherwise.
  - If the left neighbor of targeted word belongs to Measurement Gazetteers then the feature value is True and False otherwise.
  - If the right neighbor of targeted word belongs to Measurement Gazetteers then the feature value is True and False otherwise.

- Check Percent Gazetteer:
  - If the targeted word belongs to Percent Gazetteer then the feature value is True and False otherwise.
  - If the left neighbor of targeted word belongs to Percent Gazetteer then the feature value is True and False otherwise.
  - If the right neighbor of targeted word belongs to Percent Gazetteer then the feature value is True and False otherwise.

5.4.4. The feature set of the 3rd group

The feature set of the 3rd group includes the following features:

- Check POS feature flags: as described in the 1st group feature set.
- Check Gazetteers feature flags (set of nine Boolean features):
    - Check Phone Number Gazetteers:
        - If the targeted word belongs to Phone Number Gazetteers then the feature value is True and False otherwise.
        - If the left neighbor of targeted word belongs to Phone Number Gazetteers then the feature value is True and False otherwise.
        - If the right neighbor of targeted word belongs to Phone Number Gazetteers then the feature value is True and False otherwise.
    - Check ISBN Gazetteer:
        - If the targeted word belongs to ISBN gazetteer then the feature value is True and False otherwise.
        - If the left neighbor of targeted word belongs to ISBN Gazetteer then the feature value is True and False otherwise.
        - If the right neighbor of targeted word belongs to ISBN Gazetteer then the feature value is True and False otherwise.
    - Check File Name Gazetteers:
        - If the targeted word belongs to Price File Name Gazetteers then the feature value is True and False otherwise.
        - If the left neighbor of targeted word belongs to File Name Gazetteers then the feature value is True and False otherwise.
        - If the right neighbor of targeted word belongs to File Name Gazetteers then the feature value is True and False otherwise.

It worth noting that the output of the rule-based component is preprocessed using a Java program which we have developed for sentence splitting and normalization in order to produce an input that is suitable for MADA tool. Then, the output of MADA is analyzed through Java programs[3] that we have developed as well to filter the MADA output and extract relative features, and also extract the other features to eventually generate the feature set data file which represents the input dataset/text in a suitable format. The produced feature set data file is in CSV format and represents the input to the ML-based component. Figure 5.3 illustrates the feature extraction phase.

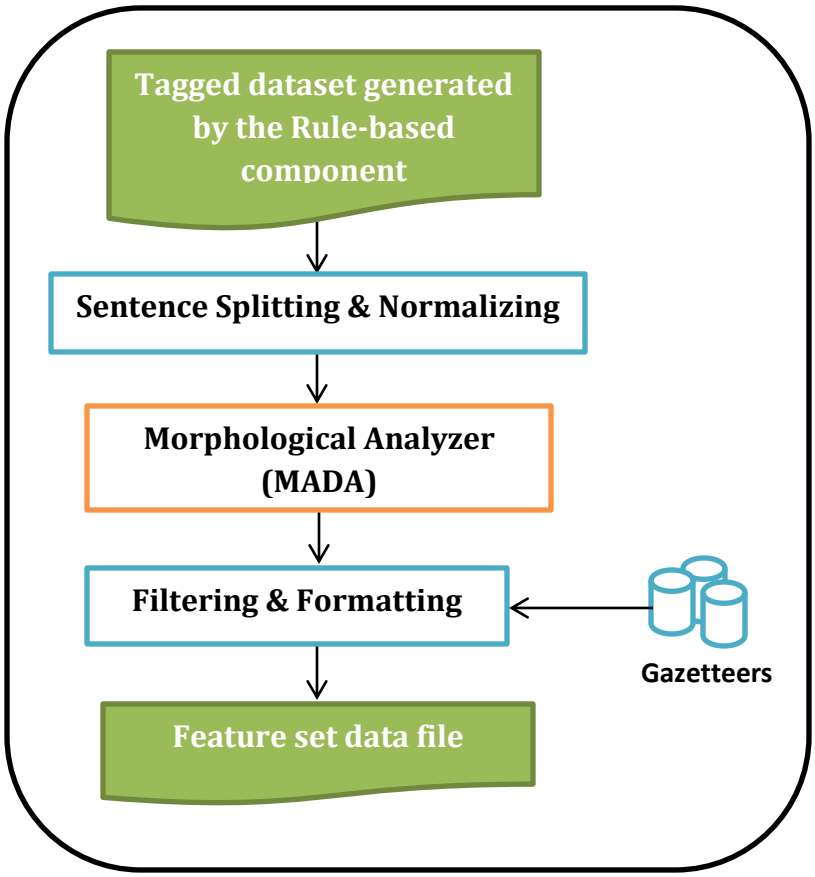

Figure 5.3: Feature Selection and Extraction Phase

[3] stanford-corenlp-1.2.0 is used as a library for the Java programs and it is available for free download at http://nlp.stanford.edu/software/stanford-corenlp-2011-09-14.tgz

5.5. Conclusion

In this research, the ML-based component depends on two main aspects. The first aspect is the machine learning approach used in training, testing and prediction phases. Three ML approaches are used and examined individually including decision trees, support vector machines and logistic regression in order to reach a conclusion with the best approach to work in our hybrid NER system for Arabic. The second aspect is the feature set which has a dedicated phase for selection and extraction. The explored features can be divided into various types of features including rule-based features, morphological features, POS features, Gazetteer features, contextual features within the Gazetteers features, and other individual contextual and word-level features. Exploring different types of features allows studying the effect of each feature type on the performance of the proposed hybrid NER system on different dimensions including named entity type and ML technique. The 11 categories of named entities that our system can extract from Arabic scripts are distributed among three groups according to their nature in which each group has a distinct set of features to be represented with. The three groups have a general set of features which represents the common features between them. The feature set has a critical impact on the performances of the ML-based component in particular and the hybrid system in general.

Chapter 6

Experimental Analysis

This chapter describes the experiments conducted to evaluate the performance of the hybrid NER system on different levels. The evaluation matrices used to measure the performance are explained in details as well. The research questions of this thesis are answered at the end of this chapter.

6.1. Confusion Matrix

The confusion matrix can be visualized as a table where the columns represent the tags predicted by an information extraction (IE) system and the rows represent the actual tags in the reference dataset. Constructing a confusion matrix is the common way exploited in evaluating IE systems and the standard IE measures, that are precision, recall and F-measure, can be extracted from the confusion matrix (Sitter et al., 2004). Figure 6.1 is a visualization of the confusion matrix.

	System Prediction		
Actual tags		+	−
	+	True Positive (TP)	False Negative (FN)
	−	False Positive (FP)	True Negative (TN)

Figure 6.1: Confusion Matrix

The standard IE measures are computed from the confusion matrix as follows:

$$Precision = \frac{True\ Positive}{True\ positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ positive + False\ Negative} \quad (2)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

The precision, as in equation (1), measures the accuracy of the system in terms of the correct classification of the recognized entities, while the recall, as in equation (2), measures the accuracy of the system in terms of the number of the entities that have been detected out of the actual number of entities in the reference dataset. Equation (3) represents the F-measure which is a harmonic mean of the precision and the recall. In this project, the results of the experiments are illustrated through these three measures.

6.2. Experimental Setup

The conducted experiments are composed of set of variables in which each experiment includes an annotated dataset which represents the reference dataset, the dataset annotated via the rule-based component, the feature set data file produced as a result of the feature extraction phase, the ML approach selected to learn the classification model, and the evaluation technique which is in our case 10-fold cross validation.

The reference datasets are the annotated datasets described in Chapter 3 with their annotation details highlighted in section 3.1.3.1 including ACE corpora (i.e. ACE 2003, ACE 2004, ACE 2005), ATB part1 v 2.0, ANERcorp and our own corpus.

The transformed datasets (as described in Chapter 3 – section 3.1.3.1) are used as inputs to the rule-based component so that second versions of these datasets are produced as outputs of the rule-based component representing **the annotated datasets**. The performance of the rule-based component, which is implemented in GATE framework, is evaluated using the built-in evaluation tool which is called AnnotationDiff.

AnnotationDiff tool enables the comparison of two sets of annotations including the one produced by the rule-based system for a dataset and the annotations of the reference dataset. The results of the evaluation process are presented with the standard IE measures (i.e. precision, recall and F-measure).

The annotated datasets produced by the rule-based component and the reference datasets (to extract the actual classes) are utilized in the feature extraction phase in order to generate **the feature set data files** that are then utilized by the ML-based component.

Three different **ML approaches** are selected to be applied to the feature set data files including decision trees, support vector machines and logistic regression approaches available in WEKA workbench via J48, Libsvm and Logistic classifiers respectively.

K-fold cross validation is an evaluation methodology used widely in the evaluation of ML-based systems. In this project, **10-fold cross validation** is chosen to avoid overfitting. 10-fold cross validation is used with each ML classifier applied to a dataset in which the dataset is split into 10 subsets in each fold where 9 subsets are used for training and the 10^{th} subset is used to test the model generated by the training phase. With 10folds, having biased resulted will be avoided as well. WEKA workbench provides the possibility of using 10-fold cross validation with each classifier and then having the results represented by the IE standard measures.

6.3. Experiments and Results

A number of experiments have been conducted in order to evaluate the performance of our hybrid NER system when applied to different datasets in order to extract different sets of named entities exploiting three different ML approaches individually.

For each dataset, three settings on the level of feature sets are examined to study their effects on the overall performance incrementally and in different combinations in order to answer the 2nd research question in this thesis. The three settings of a feature set are:

- The feature set without the rule-based features (which represents only the ML-based approach not the hybrid system)
- The feature set without the morphological features
- The feature set when all features are considered

The performance of the rule-based component is evaluated and utilized as the baseline in all experiments. Table 6.1 illustrates the results of a set of experiments carried out on ACE 2003 Newswire dataset to evaluate the performance of the system when the NE types needed to be extracted are Person, Location and Organization (i.e. 1st group):

Approach \ NE Type		Person			Location			Organization		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.66	0.8815	0.7548	0.658	0.7817	0.7145	0.4468	0.4341	0.4403
J48	Related work	0.8562	0.7668	0.809	0.8125	0.6104	0.6971	0.7821	0.5612	0.6535
J48	All Features	**0.929**	**0.934**	**0.932**	**0.87**	**0.929**	**0.899**	**0.857**	**0.627**	**0.724**
	W/O RB	0.925	0.901	0.913	0.84	0.893	0.866	0.813	0.575	0.673
	W/O MF	0.934	0.926	0.93	0.865	0.922	0.892	0.862	0.624	0.724
Libsvm	All Features	0.934	0.904	0.919	0.855	0.88	0.867	0.924	0.444	0.6
	W/O RB	0.891	0.848	0.869	0.818	0.823	0.82	0.901	0.393	0.547
	W/O MF	0.945	0.912	0.928	0.841	0.871	0.856	0.913	0.443	0.597
Logistic	All Features	0.923	0.901	0.912	0.848	0.843	0.845	0.779	0.527	0.629
	W/O RB	0.876	0.865	0.87	0.792	0.822	0.807	0.773	0.486	0.596
	W/O MF	0.916	0.891	0.903	0.839	0.823	0.831	0.78	0.506	0.614

Table 6.1: The results of the Hybrid system evaluation when applied on ACE 2003 NW dataset to recognize the 1st group NEs

Table 6.1 shows a comparison between the performance of the rule-based component, which is the baseline in the evaluation, and the performance of the hybrid system when three different classifiers (i.e. J48, Libsvm, and Logistic) are used and tested individually to compare their performance accuracy. The three settings of feature sets are tested as well in which "All Features" represents the feature set with all features are considered, "W/O RB"

represents the feature set without considering the rule-based features, and "W/O MF" represent the feature set without considering the morphological features. According to the experimental results, our feature set achieves results that outperform Abdallah, Shaalan and Shoaib (2012)'s results (i.e. Related work), the hybrid system outperforms the rule-based component and reaches the highest performance when all the features are considered and J48 (i.e. Decision Trees) is the used classifier.

Table 6.2 shows the results of a set of experiments conducted on ANERcorp dataset to evaluate the performance of the system when the NE types needed to be extracted are Person, Location and Organization. The experimental results assure the findings of the pervious set of experiments on ACE 2003 NW dataset in which our feature set achieves results that outperform Abdallah, Shaalan and Shoaib (2012)'s results (i.e. Related work), the hybrid system outperforms the rule-based system and reaches the highest performance when all the features are considered and J48 (i.e. Decision Trees) is the used classifier.

Approach \ NE Type		Person			Location			Organization		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.6581	0.7396	0.6965	0.4995	0.8259	0.6225	0.6076	0.838	0.7044
J48	**Related work**	0.949	0.9078	0.928	0.906	0.844	0.8739	0.8626	0.8599	0.8612
J48	**All Features**	**0.947**	**0.941**	**0.944**	**0.917**	**0.886**	**0.901**	**0.894**	**0.87**	**0.882**
	W/O RB	0.928	0.914	0.921	0.875	0.849	0.862	0.814	0.652	0.724
	W/O MF	0.943	0.939	0.941	0.914	0.883	0.898	0.885	0.866	0.875
Libsvm	**All Features**	0.945	0.939	0.942	0.924	0.878	0.9	0.869	0.851	0.86
	W/O RB	0.922	0.902	0.912	0.873	0.819	0.845	0.831	0.565	0.673
	W/O MF	0.936	0.942	0.939	0.92	0.878	0.898	0.855	0.852	0.853
Logistic	**All Features**	0.946	0.94	0.943	0.919	0.896	0.908	0.855	0.832	0.843
	W/O RB	0.921	0.913	0.917	0.851	0.81	0.83	0.691	0.499	0.579
	W/O MF	0.938	0.932	0.935	0.91	0.86	0.884	0.848	0.828	0.838

Table 6.2: The results of the Hybrid system evaluation when applied on ANERcorp dataset to recognize the 1st group NEs

In a comparison with the results achieved by ANERsys 1.0 (Benajiba, Rosso and Bened´I, 2007), ANERsys 2.0 (Benajiba and Rosso, 2007), ML-based NER system using CRF for Arabic (Benajiba and Rosso, 2008) and the hybrid NER system for Arabic developed by Abdallah, Shaalan and Shoaib (2012) when applied on ANERcorp, Table 6.3 illustrates the

results of these systems compared to the highest results achieved by our hybrid system. As it can be noted from Table 6.3, our hybrid system outperforms the other systems in terms of accuracy.

System \ NE Type	Person			Location			Organization		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
ANERsys 1.0	0.5421	0.4101	0.4669	0.8217	0.7842	0.8025	0.4516	0.3104	0.3679
ANERsys 2.0	0.5627	0.4856	0.5213	0.9169	0.8223	0.8671	0.4795	0.4502	0.4643
CRF-based system	0.8041	0.6742	0.7335	0.9303	0.8667	0.8974	0.8423	0.5394	0.6576
Abdallah, Shaalan and Shoaib (2012)	0.949	0.9078	0.928	0.906	0.844	0.8739	0.8626	0.8599	0.8612
Hybrid System (J48)	**0.947**	**0.941**	**0.944**	**0.917**	**0.886**	**0.901**	**0.894**	**0.87**	**0.882**

Table 6.3: The results of ANERsys 1.0, ANERsys 2.0, CRF-based system and Abdallah, Shaalan and Shoaib (2012)'s system compared to our hybrid system's highest performance when applied to ANERcorp dataset

Approach \ NE Type		Person			Location			Organization		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.6725	0.8859	0.7646	0.7344	0.6235	0.6744	0.332	0.4641	0.3871
J48	**Related work**	0.9103	0.8476	0.8778	0.8829	0.7246	0.796	0.71	0.4702	0.5657
J48	**All Features**	**0.927**	**0.881**	**0.903**	**0.897**	**0.873**	**0.885**	**0.715**	**0.571**	**0.635**
	W/O RB	0.911	0.861	0.886	0.869	0.849	0.859	0.688	0.452	0.545
	W/O MF	0.941	0.895	0.917	0.885	0.872	0.878	0.754	0.596	0.666
Libsvm	**All Features**	0.946	0.855	0.898	0.868	0.813	0.84	0.777	0.435	0.558
	W/O RB	0.901	0.793	0.844	0.818	0.785	0.801	0.867	0.144	0.247
	W/O MF	0.95	0.859	0.902	0.858	0.77	0.812	0.776	0.471	0.586
Logistic	**All Features**	0.916	0.891	0.903	0.849	0.801	0.824	0.67	0.512	0.581
	W/O RB	0.858	0.834	0.846	0.766	0.779	0.773	0.568	0.299	0.392
	W/O MF	0.919	0.875	0.896	0.84	0.756	0.796	0.691	0.521	0.594

Table 6.4: The results of the Hybrid system evaluation when applied on ACE 2003 BN dataset to recognize the 1st group NEs

Table 6.4 shows the results of a set of experiments conducted on ACE 2003 BN dataset to evaluate the performance of the system when the NE types needed to be extracted are

Person, Location and Organization. The experimental results assure the findings of the pervious set of experiments in which our feature set achieves results that outperform Abdallah et al., (2012)'s results, the hybrid system outperforms the rule-based system and reaches the highest performance when all the features are considered and J48 (i.e. Decision Trees) is the used classifier.

Table 6.5 shows the results of a set of experiments conducted on ACE 2004 NW dataset to evaluate the performance of the system when the NE types needed to be extracted are Person, Location and Organization. The experimental results show that the hybrid system outperforms the rule-based system and reaches the highest performance when all the features are considered and J48 (i.e. Decision Trees) is the used classifier.

Approach \ NE Type		Person			Location			Organization		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.3436	0.3475	0.3455	0.6349	0.7159	0.6729	0.4266	0.3473	0.3829
J48	**All Features**	**0.828**	**0.82**	**0.824**	**0.844**	**0.844**	**0.844**	**0.817**	**0.494**	**0.616**
	W/O RB	0.792	0.844	0.817	0.794	0.817	0.805	0.74	0.481	0.583
	W/O MF	0.806	0.818	0.812	0.835	0.836	0.835	0.822	0.456	0.587
Libsvm	**All Features**	0.765	0.849	0.804	0.859	0.783	0.819	0.869	0.385	0.534
	W/O RB	0.742	0.852	0.793	0.785	0.784	0.784	0.803	0.266	0.4
	W/O MF	0.8	0.81	0.805	0.848	0.76	0.801	0.857	0.383	0.529
Logistic	**All Features**	0.809	0.804	0.806	0.807	0.777	0.792	0.732	0.444	0.553
	W/O RB	0.78	0.819	0.799	0.713	0.665	0.688	0.66	0.338	0.447
	W/O MF	0.827	0.766	0.795	0.814	0.77	0.791	0.722	0.428	0.537

Table 6.5: The results of the Hybrid system evaluation when applied on ACE 2004 NW dataset to recognize the 1st group NEs

The results of applying the hybrid system on ACE 2003 NW dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.6. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier.

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.9492	0.9655	0.9573	0.8824	1	0.9375	1	1	1
J48	All Features	**1**	**0.972**	**0.986**	**0.915**	**0.956**	**0.935**	**1**	**1**	**1**
	W/O RB	0.884	0.878	0.881	0.864	0.422	0.567	0.917	0.673	0.776
	W/O MF	1	0.972	0.986	0.917	0.978	0.946	1	1	1
Libsvm	All Features	0.989	0.965	0.977	0.891	0.911	0.901	1	0.939	0.968
	W/O RB	0.868	0.822	0.844	~0	~0	~0	~0	~0	~0
	W/O MF	0.989	0.969	0.979	0.911	0.911	0.911	1	0.959	0.979
Logistic	All Features	0.986	0.986	0.986	0.875	0.933	0.903	1	1	1
	W/O RB	0.835	0.864	0.849	0.636	0.311	0.418	0.857	0.612	0.714
	W/O MF	0.979	0.986	0.983	1	0.956	0.977	1	1	1

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	1	1	1	1	1
J48	All Features	**1**	**1**	**1**	**1**	**1**	**1**
	W/O RB	0.638	0.462	0.536	0.933	0.897	0.915
	W/O MF	1	1	1	1	1	1
Libsvm	All Features	1	0.985	0.992	0.951	1	0.975
	W/O RB	~0	~0	~0	1	0.795	0.886
	W/O MF	1	0.985	0.992	0.951	1	0.975
Logistic	All Features	1	1	1	1	0.974	0.987
	W/O RB	0.487	0.292	0.365	0.945	0.885	0.914
	W/O MF	1	1	1	1	0.987	0.994

Table 6.6: The results of the Hybrid system evaluation when applied on ACE 2003 NW dataset to recognize the 2nd group NEs

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	1	1	1	1	1	1	1	1
J48	All Features	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	W/O RB	0.882	0.634	0.738	~0	~0	~0	~0	~0	~0
	W/O MF	1	1	1	1	1	1	1	1	1
Libsvm	All Features	0.973	1	0.986	~0	~0	~0	0.571	0.444	0.5
	W/O RB	0.923	0.169	0.286	~0	~0	~0	~0	~0	~0
	W/O MF	0.986	1	0.993	~0	~0	~0	0.556	0.556	0.556
Logistic	All Features	0.986	1	0.993	1	0.667	0.8	0.818	1	0.9
	W/O RB	0.754	0.606	0.672	~0	~0	~0	0.105	0.444	0.17
	W/O MF	1	1	1	0.667	0.667	0.667	1	1	1

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	1	1	1	1	1
J48	All Features	**1**	**1**	**1**	**1**	**1**	**1**
	W/O RB	0.773	0.405	0.531	~0	~0	~0
	W/O MF	1	1	1	1	1	1
Libsvm	All Features	0.816	0.952	0.879	0.6	0.333	0.429
	W/O RB	~0	~0	~0	~0	~0	~0
	W/O MF	0.952	0.952	0.952	0.545	0.667	0.6
Logistic	All Features	1	1	1	0.9	1	0.947
	W/O RB	0.5	0.429	0.462	0.25	0.556	0.345
	W/O MF	1	1	1	1	1	1

Table 6.7: The results of the Hybrid system evaluation when applied on ACE 2003 BN dataset to recognize the 2nd group NEs

The results of applying the hybrid system on ACE 2003 BN dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.7. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier. As it can be noted from Table 6.6, the rule-based component achieves the same results as the highest results achieved by the hybrid system.

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.9972	0.9916	0.9944	1	1	1	0.9898	0.9238	0.9557
J48	All Features	**0.999**	**0.992**	**0.995**	**1**	**1**	**1**	**0.994**	**0.948**	**0.97**
	W/O RB	0.871	0.281	0.424	0.778	0.364	0.496	0.938	0.554	0.696
	W/O MF	0.999	0.992	0.995	1	1	1	0.994	0.948	0.97
Libsvm	All Features	0.999	0.99	0.995	1	0.987	0.993	0.984	0.945	0.964
	W/O RB	0.775	0.252	0.38	~0	~0	~0	0.96	0.511	0.667
	W/O MF	0.997	0.99	0.994	0.987	0.961	0.974	0.984	0.942	0.962
Logistic	All Features	0.998	0.991	0.994	1	1	1	0.977	0.92	0.948
	W/O RB	0.794	0.246	0.376	0.568	0.273	0.368	0.898	0.542	0.676
	W/O MF	0.998	0.992	0.995	1	1	1	0.984	0.932	0.957

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.9423	0.9608	0.9515	0.9815	0.9815	0.9815
J48	All Features	**0.971**	**0.985**	**0.978**	**0.987**	**1**	**0.994**
	W/O RB	0.714	0.37	0.488	0.748	0.787	0.767
	W/O MF	0.971	0.985	0.978	0.987	1	0.994
Libsvm	All Features	0.963	0.963	0.963	0.987	1	0.994
	W/O RB	~0	~0	~0	0.966	0.555	0.705
	W/O MF	0.97	0.963	0.967	0.981	1	0.99
Logistic	All Features	0.955	0.948	0.952	0.987	0.987	0.987
	W/O RB	0.547	0.259	0.352	0.777	0.897	0.832
	W/O MF	0.955	0.948	0.952	0.987	0.994	0.99

Table 6.8: The results of the Hybrid system evaluation when applied on ACE 2004 NW dataset to recognize the 2nd group NEs

The results of applying the hybrid system on ACE 2004 NW dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.8. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier.

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	1	1	1	1	1	0.9722	0.9722	0.9722
J48	All Features	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0.975**	**0.987**
	W/O RB	0.822	0.634	0.716	~0	~0	~0	0.872	0.62	0.725
	W/O MF	1	1	1	1	1	1	1	0.975	0.987
Libsvm	All Features	0.97	0.997	0.983	0.818	0.6	0.692	0.991	0.917	0.953
	W/O RB	0.805	0.411	0.544	~0	~0	~0	1	0.231	0.376
	W/O MF	0.99	1	0.995	0.818	0.6	0.692	1	0.934	0.966
Logistic	All Features	1	1	1	1	0.933	0.966	0.959	0.975	0.967
	W/O RB	0.779	0.579	0.664	0.333	0.2	0.25	0.769	0.579	0.66
	W/O MF	1	1	1	1	1	1	0.952	0.975	0.963

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	0.9667	0.9831	1	1	1
J48	All Features	**1**	**0.987**	**0.994**	**1**	**1**	**1**
	W/O RB	0.731	0.241	0.362	0.984	0.579	0.729
	W/O MF	1	0.987	0.994	1	1	1
Libsvm	All Features	0.986	0.924	0.954	0.954	0.972	0.963
	W/O RB	~0	~0	~0	0.959	0.439	0.603
	W/O MF	0.974	0.949	0.962	0.955	0.981	0.968
Logistic	All Features	1	0.987	0.994	1	1	1
	W/O RB	0.643	0.342	0.446	0.849	0.579	0.689
	W/O MF	1	0.975	0.987	1	0.991	0.995

Table 6.9: The results of the Hybrid system evaluation when applied on ACE 2004 BN dataset to recognize the 2nd group NEs

The results of applying the hybrid system on ACE 2004 BN dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.9. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier.

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.9536	0.9351	0.9443	0.8571	0.9	0.878	1	0.9693	0.9844
J48	All Features	**0.996**	**0.955**	**0.975**	**0.955**	**0.94**	**0.947**	**1**	**0.978**	**0.989**
	W/O RB	0.826	0.862	0.844	0.667	0.09	0.158	0.866	0.917	0.891
	W/O MF	0.996	0.955	0.975	0.955	0.94	0.947	1	0.978	0.989
Libsvm	All Features	0.994	0.952	0.973	0.969	0.94	0.955	1	0.959	0.979
	W/O RB	0.79	0.846	0.817	~0	~0	~0	0.842	0.86	0.851
	W/O MF	0.996	0.952	0.973	0.969	0.94	0.955	1	0.961	0.98
Logistic	All Features	0.986	0.97	0.978	0.938	0.91	0.924	0.974	0.978	0.976
	W/O RB	0.779	0.704	0.739	0.56	0.209	0.304	0.835	0.865	0.85
	W/O MF	0.989	0.964	0.976	0.969	0.925	0.947	0.996	0.976	0.986

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	0.9667	0.9831	1	1	1
J48	All Features	**1**	**0.971**	**0.986**	**1**	**1**	**1**
	W/O RB	0.735	0.536	0.62	0.933	0.778	0.848
	W/O MF	1	0.971	0.986	1	1	1
Libsvm	All Features	0.985	0.943	0.964	0.991	0.981	0.986
	W/O RB	~0	~0	~0	0.977	0.778	0.866
	W/O MF	0.992	0.943	0.967	0.991	0.991	0.991
Logistic	All Features	0.986	0.971	0.978	0.991	1	0.995
	W/O RB	0.753	0.457	0.569	0.944	0.787	0.859
	W/O MF	0.993	0.964	0.978	1	1	1

Table 6.10: The results of the Hybrid system evaluation when applied on ACE 2005 NW dataset to recognize the 2nd group NEs

The results of applying the hybrid system on ACE 2005 NW dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.10. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier.

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	0.8919	0.9429	1	1	1	1	1	1
J48	All Features	**1**	**0.892**	**0.943**	**1**	**1**	**1**	**1**	**1**	**1**
	W/O RB	0.857	0.535	0.659	~0	~0	~0	0.8	0.64	0.711
	W/O MF	1	0.892	0.943	1	1	1	1	1	1
Libsvm	All Features	0.979	0.879	0.926	0.958	0.852	0.902	0.962	1	0.98
	W/O RB	0.946	0.223	0.361	~0	~0	~0	~0	~0	~0
	W/O MF	0.979	0.885	0.93	0.923	0.889	0.906	1	1	1
Logistic	All Features	0.979	0.892	0.933	1	1	1	1	1	1
	W/O RB	0.611	0.439	0.511	0.308	0.148	0.2	0.472	0.68	0.557
	W/O MF	0.993	0.892	0.94	1	1	1	1	1	1

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		1	0.9545	0.9767	1	1	1
J48	All Features	**1**	**0.983**	**0.992**	**1**	**1**	**1**
	W/O RB	0.636	0.233	0.341	~0	~0	~0
	W/O MF	1	0.983	0.992	1	1	1
Libsvm	All Features	1	0.917	0.926	0.857	0.923	0.889
	W/O RB	~0	~0	~0	~0	~0	~0
	W/O MF	1	0.9	0.947	0.857	0.923	0.889
Logistic	All Features	0.983	0.983	0.983	1	0.846	0.917
	W/O RB	0.576	0.317	0.409	0.316	0.462	0.375
	W/O MF	1	0.983	0.992	1	1	1

Table 6.11: The results of the Hybrid system evaluation when applied on ACE 2005 BN dataset to recognize the 2nd group NEs

The results of applying the hybrid system on ACE 2005 BN dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.11. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier.

Approach \ NE Type		Date			Time			Price		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.9953	0.9907	0.993	0.9756	1	0.9877	0.994	0.9881	0.991
J48	All Features	**1**	**0.995**	**0.997**	**0.985**	**1**	**0.993**	**1**	**0.991**	**0.996**
	W/O RB	0.872	0.927	0.899	0.883	0.585	0.704	0.788	0.872	0.828
	W/O MF	1	0.995	0.997	0.985	1	0.993	1	0.991	0.996
Libsvm	All Features	0.997	0.994	0.995	0.978	0.996	0.987	1	0.985	0.992
	W/O RB	0.863	0.909	0.885	0.987	0.278	0.434	0.789	0.861	0.824
	W/O MF	0.997	0.993	0.995	0.978	0.996	0.987	0.998	0.985	0.991
Logistic	All Features	0.999	0.994	0.996	0.985	0.989	0.987	0.989	0.991	0.99
	W/O RB	0.808	0.793	0.8	0.768	0.342	0.473	0.783	0.802	0.793
	W/O MF	1	0.995	0.997	0.985	0.989	0.987	0.983	0.991	0.987

Approach \ NE Type		Measurement			Percent		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		0.9326	0.9636	0.9478	0.9867	0.9867	0.9867
J48	All Features	**0.982**	**0.966**	**0.974**	**0.989**	**1**	**0.994**
	W/O RB	0.874	0.731	0.796	0.938	0.852	0.893
	W/O MF	0.982	0.966	0.974	0.989	1	0.994
Libsvm	All Features	0.982	0.96	0.971	0.994	0.977	0.986
	W/O RB	0.872	0.667	0.756	0.903	0.847	0.874
	W/O MF	0.982	0.962	0.972	0.994	0.977	0.986
Logistic	All Features	0.984	0.958	0.971	0.989	0.989	0.989
	W/O RB	0.857	0.598	0.705	0.901	0.81	0.853
	W/O MF	0.984	0.962	0.973	0.989	0.994	0.992

Table 6.12: The results of the Hybrid system evaluation when applied on ATB part1 v 2.0 dataset to recognize the 2nd group NEs

The results of applying the hybrid system on ATB Part1 v 2.0 dataset to evaluate the performance of the system when the NE types needed to be extracted are Date, Time, Price, Measurement and Percent (i.e. 2nd group) are shown in Table 6.12. According to the experimental results, the hybrid system achieves the highest performance when all features in the feature set are considered and J48 (i.e. Decision Trees) is used as the classifier.

It is worth mentioning that the hybrid system outperforms the rule-based system but at the same time their results are very close when the targeted group of NEs is the 2nd group.

In order to evaluate the hybrid system performance in recognizing Phone Number, ISBN and File Name entities (i.e. 3rd group), we had to build our own corpus that contains a representative number of entities for the 3rd group. The Experimental results show that the hybrid system achieves its highest performance when all the features in the feature set are considered (or without the morphological features) and J48 (i.e. Decision Trees) or Logistic (i.e. Logistic Regression) are used as the classifier. The results are illustrated in Table 6.13. The best performance of the hybrid system is equal to the rule-based component performance in terms of precision, recall and f-measure.

Approach \ NE Type		Phone Number			ISBN			File Name		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Rule-based system		**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
J48	**All Features**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	W/O RB	0.86	0.307	0.453	0.515	0.38	0.437	0.858	0.944	0.899
	W/O MF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Libsvm	**All Features**	1	0.993	0.996	1	1	1	1	1	1
	W/O RB	0.822	0.264	0.4	0.755	0.082	0.148	0.84	0.95	0.891
	W/O MF	1	0.993	0.996	1	1	1	1	1	1
Logistic	**All Features**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	W/O RB	0.85	0.304	0.447	0.497	0.541	0.518	0.853	0.906	0.879
	W/O MF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Table 6.13: The results of the Hybrid system evaluation when applied on our own corpus to recognize the 3rd group NEs

6.4. The Answers to Research Questions

According to the experimental results, the research questions of this thesis have been answered as follows:

- *Which approach of the rule-based, ML-based and hybrid approaches gives the best performance in recognizing named entities in Arabic scripts?*

The results show the highest performance is achieved when the proposed system which adopts the hybrid approach is used. It worth noting that the performance of the hybrid system is very close to the performance of the rule-based component alone when it comes to the numerical and temporal expressions, and the two approaches achieve the same results in recognizing NEs of the 3rd group (i.e. Phone number, ISBN and File name). As a conclusion, the hybrid approach is suitable for the three groups of NEs in which the highest performance can be achieved through the hybrid system.

- *What is the suitable feature set for Arabic NER which leads to the best performance?*

The best performance of our proposed hybrid system is achieved when all the features of different types are considered in the feature set representing the dataset.

- *Which ML approach may work effectively with a rule-based NER system to form a hybrid system for Arabic NER that improves the overall performance?*

Three ML approaches have been utilized in our proposed system individually including decision trees, support vector machines and logistic regression approaches. Decision trees approach has proved its efficiency as a classifier in the proposed hybrid system for Arabic NER in which the highest overall improvement in the performance is achieved when decision trees approach is the used classifier.

Chapter 7

Suggesting Enhancements on the Grammatical Rules

This chapter illustrates the methodology used to suggest/derive new grammatical rules in order to enhance the performance of the rule-based system. New grammatical rules that have been derived from the output of the hybrid system are described.

7.1. Methodology

In order to construct new grammatical rules that lead to improve the performance of the rule-based component in our proposed hybrid system, the output of the hybrid system is analyzed to find the weaknesses of the rules used to recognize named entities of the 1st group (i.e. Person, Location and Organization). We have developed a Java program which extracts the words that have been classified correctly by the hybrid system but misclassified by the rule-based component out of the output of the hybrid system. Table 7.1 illustrates a sample of words extracted from the output of the hybrid system when applied to ACE 2004 NW in which the class predicted by the hybrid system matches the actual class.

ID	Word in Buckwalter	Word in Arabic	Actual Class	RB Class[1]	Hybrid Class	The Sentence
1276	llArdn	للاردن	Location	Other	Location	زيارة عمل قصيرة *للاردن* يبحث خلالها
1340	krytyAn	كريتيان	Person	Other	Person	رئيس الوزراء الكندي جان *كريتيان* اليوم
1424	EAdl	عادل	Other	Person	Other	التوصل الى سلام *عادل* ودائم
10731	AlwAHdp	الواحدة	Other	Location	Other	سيكون لقاء ابناء المدينة *الواحدة* ابرز محطات مباريات غد
11731	dAnyAl	دانيال	Person	Other	Person	الى الرئيس الكيني *دانيال* اراب موي

Table 7.1: Sample of NEs correctly classified by the hybrid system but misclassified by the rule-based component separately when applied on ACE 2004 NW dataset

[1] The Rule-Based component prediction

7.2. New Grammatical Rules

The annotated dataset produced by the proposed hybrid NER system when applied to the ACE 2004 NW dataset is analyzed in order to extract new rules that can enhance the performance of the rule-based component. The new derived rules are explained below.

- The hybrid system is able to recognize full foreign Person names that may consist of two to more words. For example:

  رئيس الوزراء الكندي **جان كريتيان** (Canadian Prime Minister **Jean Chretien**)

  The rule-based component misclassified the word "كريتيان" (Chretien) as Other instead of Person. Other examples:

  أعلن فوز رئيسه **روبير غي** (Declared the winning of its President **Robeer Guy**)

  قائد الجيش الشعبي **جون قرنق** (**John Garang**, leader of the People's Army)

  The hybrid system is able to recognize full Arabian Person names that may consist of two or more words. For example:

  وتوفي **ماجد ابراهيم حوامدة** (and **Majid Ibrahim Hawamdeh** died)

  الزعيم الإسلامي السوداني **حسن الترابي** (Sudanese Islamist leader **Hassan Al-Turabi**)

  The rule-based component misclassified the words "حوامدة" (Hawamdeh) and "الترابي" (Al-Turabi) as Other instead of Person names.

  The hybrid system utilizes the POS tag "proper noun" in recognizing full Person names (second and third names in particular) taking into account presence of the preceding neighbor in the person gazetteers.

- The hybrid system classifies a word as a Location if the word's POS tag is proper noun and it belongs to Location gazetteers. For example:

  ناصر عثمان (جناح **الكويت**) (Nasser Osman (**Kuwait** Suite))

  The hybrid system classifies correctly the word "الكويت" (Kuwait) as a Location, while the rule-based component misclassified the word as Other instead of Location.

- A word is classified as a Location by the hybrid system if its POS tag is proper noun and the preceding word is proper noun and belongs to Location gazetteers. For example:

  منطقة **بيت جالا** في الضفة الغربية (The area of **Beit Jala** in the West Bank)

  The word "جالا" (Jala) is misclassified by the rule-based component, while the hybrid system classifies it correctly as Location. This rule will help classifying Location names that consist of more than one word correctly.

- The hybrid system is able to recognize Organization name which consists of two words in case the first word is a noun and belongs to Organization gazetteers, while the second word is an adjective and belongs to Organization gazetteers as well. For example:

  قال المتحدث باسم **وزارة الخارجية** (Said a spokesman for **the Ministry of Foreign Affairs**)

  The hybrid system recognizes "وزارة الخارجية" (the Ministry of Foreign Affairs) correctly as an Organization name, while the rule-based component does not recognize it as an Organization name.

With further investigation, other new grammatical rules can be derived from the outputs of the hybrid system in order to improve the performance of the rule-based system. According to our observations, the Person, Location and Organization gazetteers need to be updated and enhanced to work effectively with the grammatical rules. Also, having the datasets tagged with POS tags in advance will enable the implementation of new grammatical rules that utilize the POS information within their structure.

Chapter 8

Conclusion and Future Work

This chapter gives a conclusion to this thesis illustrating the main contributions on different dimensions. The future work is highlighted as well.

8.1. Conclusion

Named Entity Recognition (NER) is considered one of the crucial Information Extraction tasks in which many of Natural Language Processing (NLP) applications rely on as an important preprocess step. In this thesis, a hybrid system is proposed to tackle the problem of NER for Arabic. To the best of our knowledge, our system is the only hybrid NER system for Arabic that can handle extracting 11 different types of named entities including Person, Location, Organization, Date, Time, Price, Measurement, Percent, Phone Number, ISBN and File Name. This set of NE types covers the most important NE types in Arabic script and makes our system more comprehensive in the aspect of NER for Arabic.

The introduced system represents an integration between rule-based approach and Ml-based approach in order to enhance the overall performance of NER for Arabic. The rule-based component is a reproduction of a previous rule-based NER system (Shaalan and Raza, 2008) which was originally implemented within a commercial tool. Instead, GATE, a freely available developmental tool, is used to develop the rule-based component. The grammatical rules are verified and adjusted with the necessary modifications in order to enhance the performance of the NE extractors. New grammatical rules are also derived from contextual analysis of Arabic texts and added to the different NE extractors. The implemented rule-based component is capable of recognizing the 11 different types of named entities mentioned earlier. Various acquired and collected gazetteer sets are utilized

by the proposed hybrid system. Table 8.1 illustrates the number of gazetteers and rules implemented within each NE extractor in the rule-based component.

	Person	Location	Organization	Date	Time	Price	Measurement	Percent	Phone Number	File Name	ISBN	Total
Number of Gazetteers	11	20	8	12	10	8	7	3	7	3	1	**90**
Number of Rules	9	20	9	7	8	4	3	1	7	3	2	**73**

Table 8.1: The Number of Gazetteers and Rules in each NE Type

On the other hand, three different ML techniques are utilized and examined individually as part of the ML-based component including decision trees, support vector machines (SVM) and logistic regression. The first two ML techniques (i.e. Decision Trees and SVM) are known for their good performance in NER in general and Arabic NER in particular, while the third ML technique (i.e. logistic regression), to the best of our knowledge, has not been used before in Arabic NER and that gave us the opportunity to study its performance in Arabic NER through our hybrid NER system for Arabic. The ML-based component is implemented within WEKA workbench, a comprehensive tool for Data Mining and Machine Learning, where three different classifiers including J48 (i.e. an application of decision trees), Libsvm (i.e. an application of SVM), and Logistic (i.e. an application of logistic regression) are applied on different datasets for training and testing purposes.

The feature set is a very critical and important aspect in Machine Learning because ML-based models view a dataset as vectors of features in which each vector represents a word in the dataset. In this research, the explored features can be divided into various types of features including rule-based features, morphological features, POS features, Gazetteer features, contextual features and word-level features. The category of rule-based features is the main contribution of the rule-based component to the hybrid system in which the annotations produced by the rule-based component are utilized as features in the feature set. The other types of features are utilized in different combinations within the feature set in order to study their effects on the performance of the hybrid system.

The datasets utilized for training and testing are ACE 2003 dataset (NW & BN), ACE 2004 dataset (NW & BN), ACE 2005 (NW & BN), ATB Part1 v 2.0 dataset, ANERcorp dataset and our own corpus. The datasets are verified and prepared in prior of applying NER. Table 8.2 illustrates the number of annotated named entities of different types in each of the datasets used in this research.

NE type / Dataset		Person	Location	Organization	Date	Time	Price	Measurement	Percent	Phone Number	File Name	ISBN
ACE 2003	BN	711	1292	493	58	15	17	28	35			
	NW	517	1073	181	20	1	3	14	3			
ACE 2004	BN	1865	3449	1313	357	28	105	51	54			
	NW				67	4	36	30	32			
ACE 2005	BN				154	20	163	60	42			
	NW				37	7	9	22	5			
ANERcorp		3602	4425	2025								
ATB Part1 v 2.0					431	80	168	330	75			
Our own corpus										136	160	126
Total		**6695**	**10239**	**4012**	**1124**	**155**	**501**	**535**	**246**	**136**	**160**	**126**

Table 8.2: Number of different NEs annotated for NER in each dataset

A number of experiments are conducted to evaluate the performance of our hybrid NER system when applied to different datasets in order to extract different sets of named entities exploiting three different ML techniques (i.e. decision trees, SVM and logistic regression) individually. For each dataset, three settings of feature sets (i.e. the feature set without the rule-based features, the feature set without the morphological features and the feature set when all features are considered) are examined in order to study the feature set in different combination and find as a result the feature set with the best performance. According to the experimental analysis, the best performance of our proposed hybrid system is achieved when all the features of different types are considered in the feature set. Decision trees approach has proved its efficiency as a classifier in the proposed hybrid system for Arabic NER in which the highest overall improvement in the performance is

achieved when decision trees approach is used as the classifier. The experimental results show that the hybrid approach outperforms the rule-based approach and the ML-based approach separately when it comes to NER for Arabic. **This research advances the state-of-the-art of the Arabic NER** achieved by Benajiba and Rosso (2008) (i.e. CRF-based NER system) and by Abdallah, Shaalan and Shoaib (2012) (i.e. Hybrid NER system for Arabic) when applied on ANERcorp dataset. Table 8.3 illustrates the comparison between the results achieved by ANERsys 1.0 (Benajiba, Rosso and Bened´I, 2007), ANERsys 2.0 (Benajiba and Rosso, 2007), ML-based NER system using CRF for Arabic (Benajiba and Rosso, 2008) and the hybrid NER system for Arabic developed by Abdallah, Shaalan and Shoaib (2012) when applied on ANERcorp. Our hybrid NER system for Arabic outperforms the previously mentioned Arabic NER systems in terms of precision, recall and f-measure.

NE Type / System	Person			Location			Organization		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
ANERsys 1.0	0.5421	0.4101	0.4669	0.8217	0.7842	0.8025	0.4516	0.3104	0.3679
ANERsys 2.0	0.5627	0.4856	0.5213	0.9169	0.8223	0.8671	0.4795	0.4502	0.4643
CRF-based system	0.8041	0.6742	0.7335	0.9303	0.8667	0.8974	0.8423	0.5394	0.6576
Abdallah, Shaalan and Shoaib (2012)	0.949	0.9078	0.928	0.906	0.844	0.8739	0.8626	0.8599	0.8612
Our Hybrid System	**0.947**	**0.941**	**0.944**	**0.917**	**0.886**	**0.901**	**0.894**	**0.87**	**0.882**

Table 8.3: The results of ANERsys 1.0, ANERsys 2.0, CRF-based system and Abdallah, Shaalan and Shoaib (2012)'s hybrid system compared to our hybrid system's highest performance when applied to ANERcorp dataset

Four new grammatical rules are derived from the output of the hybrid system. The new rules are suggested as enhancements on the grammatical rules within the rule-based component and that automates the maintenance process of the rule-based component in which much time and efforts are saved.

8.2. Future Work

As a future work, we intend to further enhance the grammatical rules implemented within the rule-based component of our hybrid NER system especially Person, Location and Organization rules. The gazetteers are intended to be updated and enhanced as well. The analysis of the hybrid system output can help automating the process of enhancing and updating the gazetteers especially Person, Location and Organization gazetteers. We intend to continue annotating the datasets utilized in this research, e.g. labeling ACE 2004 NW dataset with Person, Location and Organization named entities. In this research, three ML approaches are utilized and examined. Thus, studying the performance of other ML approaches may help achieving better results in terms of overall accuracy. We plan to study new features in order to improve the feature set which will improve as a result the performance of the hybrid system. Further research is going to be conducted to derive more grammatical rules from the output of the hybrid system and also improve the rule extraction mechanism.

During this research, especially during the data collection phase, we have noted that the definition and specification of each category of named entities (e.g. Person, Location, Organization, Date and Time) may differ from a NER system to another. This point raises a number of issues that should be taken into consideration. One of these issues is the validity of results comparisons in which different definitions lead to different NE annotations on the same dataset. As a future work, we intend to study and analyze the impact of this point on different dimensions with the issues that rise as a result.

References

Abdallah, S., Shaalan, K. and Shoaib, M. (2012). Integrating Rule-based System with Classification for Arabic Named Entity Recognition. *Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing),* Springer-Verlag, Berlin Heidelberg, pp. 311-322.

AbdelRahman, S., Elarnaoty, M., Magdy, M. and Fahmy, A. (2010). Integrated Machine Learning Techniques for Arabic Named Entity Recognition. *International Journal of Computer Science Issues (IJCSI)*, Vol. 7, Issue 4, No 3, pp. 27-36.

Abdul-Hamid, A. and Darwish, K. (2010). Simplified Feature Set for Arabic Named Entity Recognition. *Proceedings of the 2010 Named Entities Workshop, (ACL 2010),* pp. 110-115.

Alias-i. (2008). *LingPipe 4.1.0* [online]. [Accessed 30 April 2012]. Available at: http://alias-i.com/lingpipe/

Babych, B. and Hartley, A. (2003). Improving Machine Translation Quality with Automatic Named Entity Recognition. *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT (EAMT 2003)*, pp. 1-8.

Baluja, S., Mittal, V. O. and Sukthankar, R. (2000). Applying Machine Learning For High Performance Named-Entity Extraction. *Computational Intelligence*, 16(4), pp. 586-596.

Benajiba, Y., Rosso, P. and Bened´I, J. M. (2007). ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy. *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing-2007)*, Springer-Verlag, Berlin, Heidelberg, pp. 143-153.

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. *Proceedings of Workshop on Natural Language-Independent Engineering, 3rd Indian International Conference on Artificial Intelligence (IICAI-2007)*, pp. 1814-1823.

Benajiba, Y. and Rosso, P. (2008). Arabic Named Entity Recognition using Conditional Random Fields. *Proceedings of Workshop on HLT & NLP within the Arabic World (LREC 2008).*

Benajiba, Y., Diab, M. and Rosso, P. (2008a). Arabic Named Entity Recognition: An SVM-Based Approach. *Proceedings of Arab International Conference on Information Technology (ACIT 2008),* pp. 16-18.

Benajiba, Y., Diab, M. and Rosso, P. (2008b). Arabic Named Entity Recognition Using Optimized Feature Sets. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pp. 284-293.

Benajiba, Y., Diab, M. and Rosso, P. (2009a). Arabic Named Entity Recognition: A Feature-Driven Study. *IEEE Transactions On Audio, Speech, And Language Processing*, 17(5), pp. 926-934.

Benajiba, Y., Diab, M. and Rosso, P. (2009b). Using Language Independent and Language Specific Features to Enhance Arabic Named Entity Recognition. *The International Arab Journal of Information Technology,* 6(5), pp. 464- 473.

Chieu, H. L. and Ng, H. T. (2002). Named Entity Recognition: A Maximum Entropy Approach Using Global Information. *The 19th International Conference on Computational Linguistics (COLING 2002)*

Cowie, J. and Lehnert, J. (1996). Information Extraction. *Communications of the ACM*, 39(1), pp. 80-91.

CRF++. (2012). *CRF++: Yet Another CRF toolkit* [online]. [Accessed 30 April 2012]. Available at: http://crfpp.sourceforge.net/

Diab, M. (2009). Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools*, pp. 285-288.

Eid, H. F., Salama, M. A., Hassanien, A. E. and Kim, T. (2011). Bi-Layer Behavioral-Based Feature Selection Approach for Network Intrusion Classification. *Security Technology - Communications in Computer and Information Science*, Springer, Berlin, Heidelberg, pp.195-203.

Elsebai, A., Meziane, F. and BelKredim, F. Z. (2009). A Rule Based Persons Names Arabic Extraction System. *Communications of the IBIMA,* pp. 53-59.

Farber, B., Freitag, D., Habash, N. and Rambow, O. (2008). Improving NER in Arabic Using a Morphological Tagger. *Proceedings of Workshop on HLT & NLP within the Arabic World (LREC 2008)*, pp. 2509- 2514.

Ferreira, E., Balsa, J. and Branco, A. (2007). Combining Rule-based and Statistical Methods for Named Entity Recognition in Portuguese. *V Workshop em Tecnologia da Informa¸c˜ao e da Linguagem Humana*, pp. 1615-1624.

GATE. (2012). *Overview* [online]. [Accessed 30 April 2012]. Available at: http://gate.ac.uk/

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

Habash, N., Soudi, A. and Buckwalter, T. (2007). On Arabic Transliteration. *Arabic Computational Morphology: Knowledge-based and Empirical Methods,* Springer, pp. 15-22.

Hamadene, A., Shaheen, M. and Badawy, O. (2011). ARQA: An Intelligent Arabic Question Answering System. *Proceedings of Arabic Language Technology International Conference (ALTIC 2011).*

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edition. Springer.

Lafferty, J., McCallum, A. and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*, pp. 282-289.

Maamouri, M., Bies, A., Jin, H. and Buckwalter, T. (2003). Arabic Treebank: Part 1 v 2.0. *LDC2003T06: Linguistic Data Consortium*, Philadelphia.

MADA. (2012). *MADA+TOKAN* [online]. [Accessed 30 April 2012]. Available at: http://www1.ccls.columbia.edu/MADA/index.html

Maloney, J. and Niv, M. (1998). TAGARAB: A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. *Proceedings of the Workshop on Computational Approaches to Semitic Languages (Semitic 1998)*, pp. 8-15.

Mayfield, J., McNamee, P. and Piatko, C. (2003). Named entity recognition using hundreds of thousands of features. *Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003 (CONLL 2003),* pp. 184-187.

Maynard, D., Cunningham, H., Bontcheva, K. and Dimitrov, M. (2002). Adapting a Robust Multi-genre NE System for Automatic Content Extraction. *Proceedings of the 10th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA)*, pp. 47-63.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H. and Wilks, Y. (2001) Named Entity Recognition from Diverse Text Types. *Recent Advances in Natural Language Processing 2001 Conference*.

Mesfar, S. (2007). Named Entity Recognition for Arabic Using Syntactic Grammars. *Proceedings of the 12th International Conference on Application of Natural Language to Information Systems*, Springer-Verlag, Berlin, Heidelberg, pp. 305-316.

Mitchell, A., Strassel, S., Huang, S., and Zakhary, R. (2005). ACE 2004 Multilingual Training Corpus. *Ldc2005t09: Linguistic Data Consortium*, Philadelphia.

Mitchell, A., Strassel, S., Przybocki, M., Davis, J., Doddington, G., Grishman, R., Meyers, A., Brunstein, A., Ferro, L. and Sundheim, B. (2003). Tides Extraction (ACE) 2003 Multilingual Training Data. *Ldc2004t09: Linguistic Data Consortium*, Philadelphia.

Molla, D., Zaanen, M. and Smith, D. (2006). Named Entity Recognition for Question Answering. In *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006),* pp. 51-58.

Nadeau, D. and Sekine, S. (2007). A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1), pp. 3-26.

Nguyen, H. T. and Cao, T. H. (2008). Named Entity Disambiguation: A Hybrid Statistical and Rule-Based Incremental Approach. *The Semantic Web*, Springer-Verlag, Berlin, Heidelberg, pp. 420-433.

NooJ. (2012). *NooJ* [online]. [Accessed 30 April 2012]. Available at: http://www.nooj4nlp.net

Orphanos, G., Kalles, D., Papagelis, T. and Christodoulakis, D. (1999).Decision Trees and NLP: A Case Study in POS Tagging. *Proceedings of Annual Conference on Artificial Intelligence (ACAI)*.

Paliouras, G., Karkaletsis, V., Petasis, G. and Spyropoulos, C. D. (2000). Learning Decision Trees for Named-Entity Recognition and Classification. *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*.

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. *Proceeding Conference of Association for Computational Linguistics*, pp. 426-433.

Pietra, S. D., Pietra, V. D. and Lafferty, J. (1997). Inducing Features of Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), pp. 380-393.

Riaz, K. (2010). Rule-based Named Entity Recognition in Urdu. *Proceedings of the 2010 Named Entities Workshop, ACL 2010*, pp. 126-135.

Seon, C., Ko, Y., Kim, J. and Seo, J. (2001). Named Entity Recognition using Machine Learning Methods and Pattern-Selection Rules. *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium*, pp. 229-236.

Shaalan, K. (2010). Rule-based Approach in Arabic Natural Language Processing. T*he International Journal on Information and Communication Technologies (IJICT),* 3(3), pp. 11-19*.*

Shaalan, K. and Raza, H. (2007). Person Name Entity Recognition for Arabic. *Proceedings of the 5th Workshop on Important Unresolved Matters*, pp. 17-24.

Shaalan, K. and Raza, H. (2008). Arabic Named Entity Recognition from Diverse Text Types. *Proceedings of the 6th International Conference on Natural Language Processing (GoTAL 2008)*, Springer-Verlag, Berlin, Heidel-berg, pp. 440-451.

Shaalan, K. and Raza, H. (2009). NERA: Named Entity Recognition for Arabic. *Journal of the American Society for Information Science and Technology,* 60(8), pp. 1652–1663.

Sitter, A. D., Calders, T. and Daelemans, W. (2004). *A Formal Framework for Evaluation of Information Extraction*. Technical Report, University of Antwerp, Department of Mathematics and Computer Science.

Srihari, R., Niu, C. and Li, W. (2000). A Hybrid Approach for Named Entity and Sub-type Tagging. *Proceedings of the 6th conference on Applied natural language processing (ANLC 2000)*, pp. 247-254.

Tsai, T., Wu, S., Lee, C., Shih, C. and Hsu, W. (2004). Mencius: A Chinese Named Entity Recognizer Using the Maximum Entropy-based Hybrid Model. *Computational Linguistics and Chinese Language Processing*, 9(1), pp. 65-82.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.

Walker, C., Strassel, S., Medero, J., and Maeda, K. (2006). ACE 2005 Multilingual Training Corpus. *Ldc2006t06: Linguistic Data Consortium*, Philadelphia.

YamCha. (2005). *YamCha: Yet Another Multipurpose CHunk Annotator* [online]. [Accessed 30 April 2012]. Available at: http://chasen.org/~taku/software/yamcha/

YASMET. (2002). *YASMET* [online]. [Accessed 30 April 2011]. Available at: http://www-i6.informatik.rwth-aachen.de/web/Software/YASMET.html

Zaidi, S., Laskri, M. and Abdelali, A. (2010). Arabic collocations extraction using Gate. *Proceedings of the International Conference on Machine and Web Intelligence (ICMWI 2010),* pp. 473-475.

Zhou, G. and Su, J. (2002). Named Entity Recognition using an HMM-based Chunk Tagger. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 473-480.