A Rules Based System for Named Entity Recognition in Modern Standard Arabic

Ali ELSEBAI

School of Computing, Science and Engineering
University of Salford, Salford, UK

Submitted in Partial Fulfillment of the Requirements
Of the Degree of Doctor of Philosophy
November 2009

Table of Contents

List of Tables v
List of Figures vii
Acknowledgements 2
Abstract 3
Chapter 1: Introduction and Motivation 4
1.1 Introduction 4
1.2 Motivation 7
1.3 The Methodology 8
1.4 Limitations of Current Research 9
1.5 The Contribution of the Research 11
1.6 Research Objectives 12
1.7 Thesis Structure 12
Chapter 2: Literature review and Related Work 15
2.1 Information Retrieval Vs Information Extraction 15
2.2 Information Extraction 16
2.2.1 A Historical Introduction to Information Extraction 16
2.2.2 Information Extraction Challenges 22
2.2.3 Information Extraction Tasks 24
2.2.3.1 Named Entity Recognition 24
2.2.3.2 Coreference Resolution 26
2.2.3.3 Template Element Filling 26
2.2.3.4 Scenario Template Filling 27
2.2.4 Information Extraction Approaches 28
2.2.4.1 Rule based approach 29
2.2.4.2 Rule Based system 29
2.2.4.3 Overview of Systems Adopted Rule Based Approach 31
2.2.4.4 Automatic Training Approach 32
2.2.5 Information Extraction Architecture 33
2.2.6 Information Extraction Applications 34
2.3 Arabic Natural Language Processing Challenges 37
2.4 Related Work 39
2.4.1 Systems Using Transliterated Arabic Texts 39
2.4.1.1 Rule Based Technique 40
2.4.1.2 Statistical Techniques 40
2.4.1.3 Dictionary Technique 42

2.4.2 Review of Work Using Raw Arabic Language 43
2.4.2.1 Rule Based Technique 43
2.4.2.2 Statistical Technique 45
2.4.3 Classification of Arabic Named Entity Systems 49
2.4 Summary 50
Chapter 3: Introduction to the Arabic Language 52
3.1 Introduction 52
3.2 The Morphology of the Arabic Language 54
3.3 Arabic Stemming 58
3.4 Arabic Parts of Speech 59
3.4.1 Singular, Dual and Plural in Arabic Language 61
3.4.2 Nouns and Noun phrases 62
3.4.3 The Annexation Noun 63
3.4.4 Verb Phrases 64
3.5 Summary 66
Chapter 4: The Architecture of the Arabic Information Extraction System 67
4.1 Introduction 67
4.2 The General Architecture for Text Engineering 67
4.2.1 GATE Architecture 68
4.2.2 Information Extraction System within GATE 70
4.2.3 Processing Resource 70
4.2.3.1 The Tokeniser 71
4.2.3.2 Gazetteer 72
4.2.3.3 Sentences Splitter 73
4.2.3.4 Part of Speech Tagger 73
4.2.4 Annotations 73
4.2.5 Language Resources 74
4.2 Buckwalter Arabic Morphological Analyzer 75
4.3 The Architecture of the System 76
4.4 Summary 81
Chapter 5: The Development of the Rules 82
5.1 Introduction 82
5.2 ENAMEX: Proper Name Recognition 83
5.2.1 Arabic Person Name Recognition 84
5.2.1.1 Proper Noun Preceding a Keyword 85
5.2.1.2 Proper Noun Located Next to the Keyword 86

5.2.1.3 The Proper Noun is Located away from the Keyword 88
5.2.1.4 The Morphological Analysis Stage 90
5.2.2 Organisation and Location Recognition 94
5.4 NUMEX: Percent and Money Expression Recognition 97
5.5 Summary 98
Chapter 6: System Implementation and Design 100
6.1 Introduction 100
6.2 System Design 100
6.2 The Implementation Environment 103
6.3 The System Implementation 104
6.4 Discussion of the Morphological Analyzer 110
6.5 Summary 112
Chapter 7; System Evaluation 113
7.1: Evaluation of IE Systems 113
7.2 Problem with Evaluating Arabic IE Systems 114
7.3 Evaluation Methodology 115
7.4 Comparison of our System with other System 117
7.5 Summary 119
Chapter 8: Conclusion and Future Work 121
8.1 Conclusion 121
8.2 Future Work 122
Appendix A: Qalam Transliteration Scheme 123
Appendix B: My Publications 124
Bibliography 125

List of Tables

Table 1.1: *An example of a named entity in a sentence* 5
Table 1.2: *A sample of named entities classification* 5
Table 1.3: *Importance of NE in QA applications* 6
Table 2.1: *List of the Message Understanding Conferences (Zhao, 2004)* 17
Table 2.2: *Scoring Metrics in MUC-3 (Sundheim, 1991)* 18
Table 2.3: *Tasks Evaluated in MUC-3 through MUC-7 (Chinchor, 2001)* 22
Table 2.4: *Word derivation from the root "ktb"* 38
Table 2.5: *An example showing a proper noun after a keyword (Benajiba and Rosso, 2008)* 41
Table 2.6: *Non named entity words starting with a capital letter.* 42
Table 2.7: *An example of an Arabic phrase* 45
Table 2.8: *Incorrect tagging of the word Salem* 46
Table 2.9: *An example showing an incorrect tag for the word Dubai* 47
Table 2.10: *Classification of some systems depending on the approach and the text applied* 49
Table 3.1: *The variation of the letter (gh).* 54
Table 3.2: *Morphological analysis of the word قبل* 55
Table 3.3: *A sample of Arabic affixes* 55
Table 3.4: *An agglutinated form of the Arabic word "ليفاوضونهم, to negotiate with them"* 56
Table 3.5: *Root extraction from the stem "enter, أدخل"* 56
Table 3.6: *A sample of words extracted from the root (write/كتب).* 56
Table 3.7: *Part-of-Speech tagging for the sentence كتب فريد قبل وسيم* 60
Table 3.8: *The position of the dual in Arabic* 61
Table 3.9: *The position of the plural in Arabic* 62
Table 3.10: *An example of an irregular plural* 62
Table 3.11: *Example of the differentiation between the masculine and feminine.* 62
Table 3.12: *An example of a noun phrase.* 63
Table 3.13: *Examples of noun phrases ending with a noun.* 63
Table 3.14: *Examples of named entities obtained by annexation.* 64
Table 3.15: *Example of a verb phrase where the verb is in the normal position.* 64
Table 3.16: *An example where the subject is omitted.* 65
Table 3.17: *An example where the object comes before the subject* 65
Table 4.1: *Examples of introductory verbs* 77
Table 4.2: *Examples of introductory words (IWL)* 78
Table 4.3: *Examples of place names* 78
Table 4.4: *Examples of stop words.* 79
Table 4.5: *Example of conjunctions* 79
Table 4.6: *Example of country names* 80
Table 4.7: *Example of organization names* 80
Table 4.8: *Examples of Arabic person names starting with "AAL"* 81
Table 5.1: *Examples of clues associated with some named entities* 84
Table 5.2: *An example showing the position of a proper name with regards to its keyword.* 84
Table 5.3: *An example of a proper noun preceding a keyword* 85

Table 5.4: *An example showing a stop word preceding a keyword* 86
Table 5.5: *Example showing a stop word next to a keyword* 86
Table 5.6: *An example showing an IVL word next to a keyword* 87
Table 5.7: *An example showing an IVL word next to a keyword* 88
Table 5.8: *An example illustrating an adjective between a keyword and a proper noun* ... 88
Table 5.9: *An example showing a proper noun starting with definite article (ال)* 90
Table 5.10: *An example showing a conjunction followed by the city name.* 90
Table 5.11: *An example showing an adjective followed by a proper noun.* 91
Table 5.12: *An example showing a place named entity.* 92
Table 5.13: *An example where the proper noun is missed by BAMA.* 93
Table 5.14: *illustrate the variation of the name of the organisation in Arabic text* 95
Table 5.15: *An example of name variation in Arabic location names.* 96
Table 5.16: *The different months' names as used by different countries.* 97
Table 5.17: *The days of the week in the Arabic language* 97
Table 5.18: *The digital and alphabetic numbers in Arabic* 98
Table 5.19: *A sample of currency names in Arabic* 98
Table 6.1: *the output of the experiment given in figure 6.13* 110
Table 7.1: *Examples of person names appearing next to a trigger word* 115
Table 7.2: *The initial results obtained by our system* 115
Table 7.3: *Examples showing stop words next to a trigger word* 116
Table 7.4: *System performance after using stop words.* 116
Table 7.5: *Example of words begining with (ال, the)* 117
Table 7.6: *System performance after considering words beginning with (ال, the) isolated* 117
Table 7.7: *Comparison of Mesfar's results and our results (in bold and italic font)* 119
Table 7.8: *Summary of the performances of Arabic NER systems* 120

List of Figures

Figure 2.1: *An Example of a Text (Cunningham, 1999)* 24
Figure 2.2: *Named entity recognition in GATE (Cunningham, 1999)* 25
Figure 2.3: *Coreference resolution in GATE (Cunningham, 1999)* 26
Figure 2.4: *Template elements in GATE (Cunningham, 1999)* 27
Figure 2.5: *Scenario template in GATE (Cunningham, 1999)* 28
Figure 2.6: *Proteus system architecture. (Kaiser and Miksch, 2005)* 31
Figure 2.7: *IdentiFinder system architecture. (Weischedel et al. 1996)* 33
Figure 2.8: *The architecture of information extraction systems (Kaiser and Miksch, 2005)* 34
Figure 2.9: *EMPathIE system modules within GATE (Humphreys, 2000)* 36
Figure 2.10: *FACILE system architecture (Black, 1998)* 37
Figure 2.11 : *Examples of a proper noun next to a trigger word (Pouliquen et al., 2005)* 42
Figure 2.12: *The architecture of the PERA System (Shaalan and Raza, 2007)* 44
Figure 2.13: *Different forms of Arabic phrases* 45
Figure 2.14: *The architecture of Shamsi and Guessoum's system (Shamsi and Guessoum, 2006)* 46
Figure 2.15: *The output of Buckwalter for the word (Salem, سالم).* 47
Figure 2.16: *The output of Buckwalter for the word (Dubai, دبي).* 48
Figure 4.1: *The GATE Architecture (Cunningham, 2000)* 68
Figure 4.2: *The GATE graphical interface* 69
Figure 4.3: *The workflow of ANNIE (Cunningham et al, 2010)* 71
Figure 4.4: *An example of a text after tokenization.* 72
Figure 4.5: *Various annotations of an Arabic text.* 74
Figure 4.6: *Buckwalter output for the word (Kateb, كاتب).* 76
Figure 4.7: *The system architecture and the flowchart* 77
Figure 5.1: *The first proper noun rule* 85
Figure 5.2: *Proper noun rule when a stop word is next to a keyword* 86
Figure 5.3: *The proper noun rule when an IVL word is next to a keyword* 87
Figure 5.4: *The proper noun rule when an IWL word is next to a keyword* 87
Figure 5.5: *Proper noun rules for nouns starting with (ال, the) or when a conjunction is followed by a country or city name* 89
Figure 5.6: *Rule for selecting a proper name from BAMA* 91
Figure 5.7: *Buckwalter output for the word (نوري, Nouri)* 92
Figure 5.8: *Buckwalter output for the word (دبي, Dubai)* 93
Figure 5.9: *Buckwalter output for the word (لارجاني, Larigany)* 94
Figure 6.1: *class diagram for our system.* 101
Figure 6.2: *initializing GATE within ANEFinder class* 101
Figure 6.3: *The JAPE rule for IVL* 102
Figure 6.4: *initializing BAMA within NamedEntity class and testing the word by BAMA.* 103
Figure 6.5: *The workspace of our application* 105
Figure 6.6: *illustrate the creation of our project* 105
Figure 6.7: *illustrate the selection of the appropriate encoding* 106
Figure 6.8: *illustrate the integration of the systems with eclipse via JAVA library* 107
Figure 6.9: *illustrate the importation of GATE within eclipse* 108
Figure 6.10: *illustrate the importation of Buckwlter (BAMA) within eclipse* 108

Figure 6.11: *illustrate the importation of our JAVA classes within eclipse*..... 108
Figure 6.12: *illustrate GATE, BAMA, and our JAVA classes in our project*..... 109
Figure 6.13: *the experiment over one article.*..... 109
Figure 6.14: *BAMA incorrect stemmer solutions of the word (المالكي, Almalikiy)*..... 111
Figure 7.1: *Mesfar's system architecture (Mesfar, 2007).*..... 118

Acknowledgements

I wish to thank Dr. Farid Meziane, my thesis supervisor, for his guidance and continuous encouragement during this research. And I would like to thank my brother Al-Mabrouk for his support and care.

Abstract

The amount of textual information available electronically has made it difficult for many users to find and access the right information within acceptable time. Research communities in the natural language processing (NLP) field are developing tools and techniques to alleviate these problems and help users in exploiting these vast resources. These techniques include Information Retrieval (IR) and Information Extraction (IE). The work described in this thesis concerns IE and more specifically, named entity extraction in Arabic. The Arabic language is of significant interest to the NLP community mainly due to its political and economic significance, but also due to its interesting characteristics.

Text usually contains all kinds of names such as person names, company names, city and country names, sports teams, chemicals and lots of other names from specific domains. These names are called Named Entities (NE) and Named Entity Recognition (NER), one of the main tasks of IE systems, seeks to locate and classify automatically these names into predefined categories. NER systems are developed for different applications and can be beneficial to other information management technologies as it can be built over an IR system or can be used as the base module of a Data Mining application. In this thesis we propose an efficient and effective framework for extracting Arabic NEs from text using a rule based approach. Our approach makes use of Arabic contextual and morphological information to extract named entities. The context is represented by means of words that are used as clues for each named entity type. Morphological information is used to detect the part of speech of each word given to the morphological analyzer. Subsequently we developed and implemented our rules in order to recognise each position of the named entity. Finally, our system implementation, evaluation metrics and experimental results are presented.

Chapter 1

Introduction and Motivation

1.1 Introduction

Natural Language Processing (NLP) is the field of Artificial Intelligence (AI) that attempts to understand and process natural languages in the way humans do it. Hence, the main task of NLP is centered on designing and building software that can analyze, understand, and generate natural language (NL). To understand a language, the knowledge of words alone is not enough. One has to understand the meaning of the words, how they are pronounced and how the words combine to form sentences (Meziane, 1994). Allen (1987) identified six types of knowledge that are essential to understand a language namely: phonetic and phonological knowledge, morphological, syntactic, semantic, pragmatic and some general knowledge.

There are many applications developed in the past few decades in the NLP field. These include Information Retrieval (IR), Machine Translation (MT), Data Mining (DM), Question Answering (QA) and Information Extraction (IE). IE is the practical way to get one step closer to the goal of NLP. In a narrow sense, IE reduces facts in text into a structured representation (Zhao, 2004). However, the IE expression is an umbrella that covers a number of tasks which have been identified by the Message Understanding Conference (MUC). The basic and essential task for IE or NLP in general is named entity recognition (NER). The term "Named Entity", now widely used in NLP, was coined for the first time in the Sixth MUC conference (MUC-6) (Grishman and Sundheim, 1996).

However, text usually contains all kinds of names, for example person names, company names, city names, country names, sports teams, chemicals and lots of other names from a specific domain. NER seeks to locate and classify automatically these names in text into predefined categories. In fact, NER has two main sub-problems. One is NE detection (NED) that is the identification of the portion of text that forms a NE, and the second is NE classification (NEC), the process of assigning a category label to the identified span of text.

Let us consider for instance the example given in Table 1.1.

Table 1.1: An example of a named entity in a sentence

خالد سافر الإثنين الماضي من عمان ليلتقي بأحمد في القاهرة
Khaled traveled last Monday from Amman to meet Ahmed in Cairo

There are five named entities in the above Arabic sentence, the named entity extractor would identify these entities and label them as shown in Table 1.2.

Table 02: A sample of named entities classification

The Named Entity	***The Type***
(خالد, Khaled)	Person
(الإثنين,Monday)	Date
(عمان, Amman)	Location
(أحمد, Ahmed)	Person
(القاهرة,Cairo)	Location

Extracting named entities benefits many NLP tasks such as question answering (QA) systems, one of the most complicated NLP tasks because it attempts to provide accurate answers to specific user queries. In order to answer questions like ''who is the head of the British Government?'', it is useful to know that the expected answer is the name of a person, and obviously, to consider as candidate answers only entities of this type. Consequently, a QA system does not only retrieve the relevant documents (as is the case in an IR system) but extracts the answer and replies to the question automatically. Although

most search engines allow users to form some types of queries using wild cards for pattern matching, they still do not help very much in locating the targeted NE. Furthermore, Srihari and Li (2000) stated that out of the 200 questions that comprised the Text REtrieval Conference (TREC-8) Question Answering track competition, over 80% asked for an NE, e.g. *who* (PERSON), *when* (TIME[DATE), *where* (LOCATION), *how far* (LENGTH). The beginning of the questions given in Table 1.3 usually requires an answer containing a NE. On the other hand, NE can help narrow down the targeted text portions which contain potential answers. Thus, applying a NE tagger to a QA has been proven to be very helpful. Many studies show that the precision of a QA system relies significantly on the performance of the NER system included within. For instance Greenwood and Gaizauskas (2007) applied a NER system in order to improve the performance of an answer extraction module. They reported that the accuracy has improved in the answering questions systems.

Table 1.3: Importance of NE in QA applications

	The question	***Referred to***
1	Who/Whom	Person
2	When	Time/Date
3	Where/What place	Location
4	What time (of day)	Time
5	What day (of the week)	Day
6	what/which month	Month
7	What	Name
8	How rich	Money
9	How money	Number
10	How long	Duration

In information retrieval, which aims at retrieving relevant document for user formulated queries, usually in natural language, Thompson and Dozier (1997) reported

that 67.83%, 83.4% and 38.8% of the queries contained one or more Named Entities (NEs) when used to extract information from the Wall Street Journal, Los Angeles Times and Washington Post respectively. Hence, an improvement of the retrieval of documents for queries containing NEs would boost significantly the performance of the global IR systems.

In Machine Translation (MT), the correct identification of named entities (NEs) is an important problem. Incorrectly translating NEs as common nouns leads to ambiguities or obligates extensive post-editing. Often failure to precisely identify NEs has an effect, not only on a local and immediate context, but also on the global syntactic and lexical structure of the translation. The post-editing step is more expensive when the errors of a MT system are mainly in NEs translation. For these reasons Babych and Hartley (2003) conducted research where they tagged a text with the NER module of Sheffield's GATE information extraction (IE) system (Cunningham, 2000) as a pre-processing step of MT. They reported that they have reached a higher accuracy with this new approach, and they indicate that specific components of IE technology could boost the performance of current MT systems.

Moreover, NE recognition can enhance data mining tasks, and cross language IR. For instance; it was shown in (Larkey and AbdulJaleel, 2003) that in Arabic-English cross language IR, the performance degrades seriously when the bilingual dictionaries do not contain Arabic proper names. Hence, NE extraction tools are essential for many applications in NLP.

1.2 Motivation

The work described in this thesis concerns IE and more specifically, named entity extraction in Arabic. In some languages such as English and French, capitalization is a

good clue to identify proper names, however, in other languages such as Arabic this feature does not exist. Thus we cannot mark the names in the text by simply looking at the first letter of the word and this increases the difficulty of extracting named entities in Arabic texts. It has been shown that the absence of capital letters in the Arabic language is the main obstacle to obtain high performance in NER systems (Benajiba et al., 2007). However, with the huge amount of published data in Arabic, over 200,000 websites and 300,000 users over the net (Mesfar, 2007), we recognize that developing a system to extract important data from documents becomes essential. The Arabic language is a language of significant interest to the NLP community mainly due to its political and economical significance, but also due to its interesting characteristics (Benajiba and Rosso., 2008). In this thesis we aim to develop a rule-based Arabic NER system, in this thesis, we use rule based system in the context of information extraction rather than in the context of knowledge based system. A rule based system uses rules to extract information. There is little research on Arabic NER (Maloney et al, 1998; Samy, 2005; Abuleil, 2004; Harmain and Aljohar, 2001, Shaalan and Raza, 2009).

1.3 The Methodology

In this thesis we describe a method to recognize Arabic named entities in Arabic newspapers. Initially, we collected one hundred articles (test corpora) from Aljazeera website (Aljazeera, 2008). Then manually, we annotated and highlighted the entire named entities in the test corpora. Consequently and from our observation, we identified several rules to specify these names; afterwards we developed these rules and implemented them using the JAVA programming language. In addition, in order to tag named entities in Arabic texts, lists of trigger words have been identified to help identify the position of the proper names in the text. By using keywords we mark noun phrases that might include a

certain name then we process these phrases to extract these names. However, we selected another hundred articles, from the same website, and we run our system over these articles. The results were satisfactory and encouraging despite the experiment being at an initial stage. However, we reviewed the test corpora one more time and consequently added more rules to our system and improved the existing rules and the precision of our system. We choose another hundred articles and we run our system over the new articles. As expected, our system achieved better results than the first attempt. We reiterated this process until we analyzed one thousand articles and we marked and extracted the named entities from these articles, then we classified into their categories i.e. person name, city name, organization name, date etc. We have developed our IE system using the General Architecture for Text Engineering (GATE) system. GATE comes with a default information extraction system called ANNIE (A Nearly-New IE system) (Maynard et al, 2001). GATE is the most freely available advanced NLP tool that many researchers are using around the world (Cunningham, 2000). In addition, in the design and implementation of our system, Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004) is built over GATE to achieve our goals. BAMA is widespread and heavily used in the literature; it has been used for example in the Language Data Consortium (LDC) Arabic POS tagger, Peen Arabic Treebank, and the Prague Arabic Dependency Treebank (Attia, 2006).

1.4 Limitations of Current Research

The research presented in this thesis is conducted for specific corpora, using a specific methodology. Hence it is not attempting to solve all the issues related to NER in Arabic.

To clarify the scope of this research, its limitations can be summarised as follows:

- The entire experiment relay on a specific domain, the political domain, hence the techniques and rules developed for the current system are restricted to this domain

and will probably yield to lower precision if applied to general or different domains.

- Shaalan and Raza (2009) carried out an extensive research on Arabic language NLP tools and resources in general (e.g., corpora, gazetteers, POS taggers, etc.). They concluded that in comparison with other languages, Arabic lacks mature linguistic resources, especially free resources available for research purposes. Consequently we constructed our own corpora based on resources we have access to. This was a laborious and time consuming process and we limited our corpora to satisfy the requirements of our system.
- Most researchers on Arabic language do not keep the corpora they have designed after conducting their experiments or the free resources they have used to evaluate their system are no more available. For example, the corpus used by Mesfar (2007) is no more available as the free access to the "Le Monde diplomatic" Arabic articles is removed. This makes the comparison of the various systems nearly impossible and we have experienced this problem in our study. This will be explained in Chapter 7.
- Based on the limitations of the resources for the Arabic language it is difficult to make any comparison with NER systems developed for other languages.
- There are two approaches for information extraction, the rule based approach (also called the knowledge engineering approach) and the statistical approach (also called the training approach). We have used a rule based approach and its justification will be given in chapter 5.
- Our system is mainly developed for Modern Standard Arabic.

1.5 The Contribution of the Research

The work developed and reported in this thesis makes many contributions to the field of NER for the Arabic language. These can be summarised as follows:

- Given the characteristics of the Arabic language, NLP systems developed for the Arabic language make heavy use of morphological analysers and part of speech taggers regardless of the approach used for information extraction. The precision of the developed systems are strongly linked to the accuracy of the morphological analysers and part of speech taggers. In many systems, the first outputs of the morphological analysers and part of speech taggers are taken as the correct solution (Shamsi and Guessoum, 2006). However, this is not always true. In our approach, we have developed rules that scan all the solutions provided and select the correct one.
- We adopted a systematic way in developing our system, following an object-oriented approach that made use of well know systems in the NLP field namely GATE and BAMA. This allowed us to develop a reliable and reusable system that can benefit the Arabic information processing community.
- We conducted a thorough evaluation of our system and the results obtained are very promising.
- In our application we have used Buckwalter's stemmer which is known to return sometimes incorrect or incomplete result. We have developed rules to identify these cases as we will illustrate this in chapter 6. Realistic outstanding
- The lack of free resources has been identified as the main factor hindering the development of Arabic NLP systems. Hence we created our own corpora and we

are planning to make our corpora freely available on a website to allow other researchers to use it to compare their systems with ours.

- Our research has been published in many reputable conferences. A list of our publications is given in appendix B.

1.6 Research Objectives

The main objectives of this research are summarised as follows:

- Develop several rules to identify the Arabic named entities
- Identify the main aspects and features of named entities in Arabic.
- Thoroughly study the development and implementations of NE recognition systems in other languages and identify components and approaches that can be reused for the Arabic language.
- Study the computational issues involving the identification of Arabic Names in Arabic documents.
- Implement a prototype system to test the findings of this research and evaluate the system.
- Compare the performance of our system to those of similar systems.

1.7 Thesis Structure

This section describes the structure of this thesis:

Chapter 1 introduces the research topic, and presents the aim, objectives and the methodology adopted for developing our research. We highlighted the limitations of our approach and the contribution we have made to the field of Named Entity Recognition in the Arabic language.

Chapter 2 covers the background on Information Retrieval (IR) in general and Information Extraction (IE) in particular. We started by providing a short history of the development of the IE field and the role played by the Message Understanding Conferences (MUCs) in the development of the field. We also described the two approaches used in IE, namely rule based (Knowledge Engineering) and the automatic training (statistical) approaches. The chapter then provides some examples of the fields where IE is applied. We concluded by addressing the challenges in the Arabic language.

Chapter 3 covers some of the characteristic of the Arabic language relevant to the current research, we have illustrated that the Arabic language has a rich vocabulary and a complex morphology. We reviewed several approaches and systems used for Arabic morphological analyses and systems used for Arabic parts of speech identification.

Chapter 4 describes the language engineering tools used in our research. We first describe GATE and its three subsystems namely the GATE Document Manager, GATE Graphical Interface and CREOLE (a Collection of REusable Objects for Language Engineering). Then we describe the Buckwalter Arabic Morphological Analyser (BAMA). We concluded the chapter by describing the architecture of our system and how GATE and BAMA have been used in our architecture.

In chapter 5 we summarised the characteristics of the Arabic language and the approaches used in the field of the NE recognition and justified our choice for a rules based system for our Arabic NER. In this chapter we have also developed the rules for our NER system.

In chapter 6, we describe the environment used for the implementation of our architecture. This was followed by the evaluation of our system and highlighted the difficulties encountered in evaluation Arabic IE systems in general.

In chapter 7, we reviewed several Arabic named entity recognition system and summarised the methodologies they have used, the corpora used for their evaluation and the evaluation of these systems. We conclude the chapter by providing a comparison of the results obtained by our system with Mesfar's system (Mesfar, 2007). It would have been interesting to compare our results with those obtained by the PERA system, but unfortunately we did not have access to the corpora used by PERA although we contacted the author several times.

In chapter 8, we conclude the research developed in this thesis and summarise the main findings and highlighted possible future development of the current research.

Chapter 2

Literature Review and Related Work

There has been an explosive growth in the amount of information available on networked computers around the word, much of it in the form of natural language (NL) documents which are mainly analyzed by natural language processing (NLP) techniques. The most important of these techniques are Information Retrieval (IR), Information Extraction (IE), Machine Translation (MT), Question Answering (QA) and more recently Data Mining (DM) and text mining. This chapter reviews some of the areas and events in the NLP field that lead to the development of the named entity recognition field.

2.1 Information Retrieval Vs Information Extraction

Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in a storage containing information useful to him (Hiemstra, 2001). IR systems are software systems that are capable of retrieving a set of electronically stored documents based on users' queries. The retrieved documents are known as the relevant documents. On the other hand IE systems retrieve specific information from documents; and the IE systems would mark the documents that contained key terms such as "terrorist" or "bomb" after which information analysts would examine these documents (Jackson, 2002). Unlike IR, IE involves shallow parsing of text, such as part-of-speech (PoS) tagging and text chunking. Indeed, research has shown that shallow parsing is sufficient for tasks such as IE

or question answering (Buchholz, 2002). IR and IE are complementary, and when combined they can provide powerful tools for NLP (Eikvil, 1999).

2.2 Information Extraction

The huge amount of data available nowadays has created a new challenge. The more data is available on electronic format, the more difficult it is to find and extract specific facts, since the data is unstructured and the majority of it is available in a human language form. Moreover, much time is spent in reading and analysing the obtained documents in order to extract the required information. On the other hand, the main objective of NLP is to understand the implicit meaning of texts and transform them into a machine intelligible representation. Then the computational power of machines would enable us to process the information in more simple ways, such as producing answering questions. Thus, it is possible to represent the underlying world knowledge in a simple format like templates. Hence, IE is a realistic way to accomplish NLP goals. Therefore, focus has shifted in the last two decades to IE; the task of which is to locate the desired information (DARPA, 1998). IE has well defined tasks, uses real-world texts, and poses complicated and motivating NLP problems. IE tasks are capable of performing at the same level as humans and with high performance; accordingly IE is attractive from the NLP perspective (Cowie and Lehnert, 1996). It is domain-dependent, in a narrow sense; IE reduces facts in text into a structured representation (Grishman, 1997) such as tables and templates.

2.2.1 A Historical Introduction to Information Extraction

IE as a research field of its own, has been introduced by Harris in the 1950's (Harris, 1957). However, the first applications have been reported much later in the 1980's within the medical domain (Sager et al, 1987). Since then, significant work has been carried out in

various domains, but the field has been heavily influenced by the Message Understanding Conferences (MUCs) funded by the U.S Government to establish the tasks of IE and benchmark corpora (Grishman and Sundheim, 1996). Altogether, there were seven MUC assessments during a ten year span, starting in 1987. Table 2.1 summarises the year and the domain on interest of each MUC.

Table 2.1: List of the Message Understanding Conferences (Zhao, 2004).

Project	***Year***	***Domain***
MUC-1	1987	Navy Operations
MUC-2	1989	Navy Operations
MUC-3	1991	Terrorism in Latin America
MUC-4	1992	Terrorism in Latin America
MUC-5	1993	Corporate Joint Venture and Microelectronics
MUC-6	1995	Corporate Management Succession
MUC-7	1998	Airplane Crashes/Rocket Launches

A brief chronological description of the MUCs is as follows:

- ***MUC-1 (1987):*** Researchers from six organizations ran their systems on the test data during the conference, then demonstrated and explained how their systems analyzed texts. Ten narrative paragraphs from naval messages had been used as a training corpus and two others as test data. However, there was no task definition and there were no evaluation criteria.
- ***MUC-2(1989):*** Eight systems were used with a training corpus of 105 texts. Again the domain was naval message narratives of four different types, a dry-run test set of 20 narratives, and a final test set of five. An IE task was specified which concerns the identification of ten different pieces of information representing them as slot fillers in a *template* emulating a semantic frame. In addition, assessment

criteria were defined, but by consent deemed not to have been sufficient and systems were scored by hand as opposed to a hand-generated answer key.

- ***MUC-3 (1991):*** One of the fundamental differences between MUC-3 and MUC-4 and previous conferences is in their choice of texts. MUC-3 and MUC-4 made use of news articles on the subject of Latin American terrorism, whereas the previous conferences made use of naval tactical message narratives (Sundheim, 1991). The planning for MUC-3 began while MUC-2 was still in progress, with suggestions from MUC-2 participants for improvements. Fifteen systems participated and a corpus of 1,400 texts on the subject of Latin American terrorism was used that included 16 text types (transcribed speeches, newspaper articles, editorial reports, etc.). The template developed contained slots for 17 pieces of information. The scoring metrics were refined and implemented in a semi automated scoring system. Formal evaluation criteria adapted from notions developed in IR (particularly, precision and recall) were introduced, as shown in Table 2.2. A semi-automated scoring system was developed and made available for use by the participants during development, and significant progress was made.

Table 2.2: Scoring Metrics in MUC-3 (Sundheim, 1991)

Measure	***Definition***
Recall	$\frac{\text{\# correct fills generated}}{\text{\# fills in key}}$
Precision	$\frac{\text{\# correct fills generated}}{\text{\#fills in generated}}$
Overgeneration	$\frac{\text{\# spurious fills generated}}{\text{\#fills generated}}$
Fallout	$\frac{\text{\# incorrect+ spurious generated}}{\text{\# possible incorrect fills}}$

The two primary measures of performance were completeness (recall) and accuracy (precision). Recall a measure of the percentage of information available which is

actually found and precision is a measure of the percentage correctness of the information produced. There were two additional measures, one to isolate the amount of spurious data generated (overgeneration) and the other to determine the rate of incorrect generation as a function of the number of opportunities to incorrectly generate (fallout) (Sundheim, 1991).

- ***MUC-4 (1992):*** Seventeen sites participated. The domain was again Latin American terrorism, the number of information bearing slots increased from 17 in MUC-3 to 22. Changes were made to the task definition, corpus, measures of performance and test protocols in order to:

  - Provide greater focus on spurious data generation.

    A few changes were made to the template scoring software to make the generation of spurious data more apparent. One of these changes focuses on overgeneration at the slot level (generating more slot values than were expected), while others focus attention on overgeneration at the template level (generating more templates than were expected). To address the spurious slot-value issue, an additional method of assessing penalties for missing and spurious data (called the "Matched/Spurious" method) was incorporated, completing the picture provided by the three measures that have been developed for MUC-3. To address the spurious template issue, a preliminary step in the alignment of response templates with key templates was implemented that requires that minimal "content-based mapping conditions" be met in order for alignment to occur (Sundheim, 1992).

  - Better assessment of system independence from training data.

Reuse for MUC-4 of the same domain and fundamentally the same task as used for MUC-3 raised the concern that the "generality" of the systems would come into question. To address these concerns, a controlled generality test was added to the test protocol.

- Make scoring more consistent.

  The scoring program was updated to further automate the scoring of set-fill slots. It was updated to score some string fills automatically. The coverage of the interactive scoring guidelines was extended. These updates were meant to ensure greater consistency in template scoring among people and across scoring runs.

- Provide means for more valid score comparison between systems.

  Two innovations in the area of scoring were made to address these issues. First, a scientifically sound, single-score measure was incorporated that enabled systems to be ranked. This measure, known as the F-measure, allows different weightings of recall and precision. Second, a method of doing statistical significance testing was incorporated into the test protocol.

The MUC-4 featured an enhanced evaluation methodology, greater participation, and significantly more conclusive results than those recorded in the MUC-3 proceedings (Sundheim, 1992). In summary, MUC-3 and MUC-4 offered benchmarks for the field of NLP in general and IE technology in particular (Sundheim, 1992).

- ***MUC-5(1993):*** Seventeen systems participated (fourteen American, one British, one Canadian and one Japanese, this marked the first non-US involvement). Significant auxiliary resources were supplied. Development and test corpora sizes

were increased. Scoring was modified to include new evaluation metrics and the scoring program enhanced. However, the metrics used for MUC-5 evaluation represent a major update to those used for MUC-4. The official MUC-5 metrics express error rates while the official MUC-4 metrics express performance in terms of recall and precision (Chinchor, 1993).

- ***MUC-6 (1995):*** Grishman (1996) identified the goals of the MUC-6 as follows:
  - Demonstrating task-independent component technologies of IE which would be immediately useful.
  - Encouraging work to make information extraction systems more portable.
  - Encouraging work on "deeper understanding."

However, seventeen sites overall took part and four subtasks were established. Participants were invited to enter their systems in as many as four different task-oriented evaluations. The Named Entity and Coreference tasks entailed Standard Generalized Markup Language (SGML) annotation of texts and were being conducted for the first time. The other two tasks, Template Element and Scenario Template, were IE tasks that followed on from previous MUC evaluations. The domain of the scenario extraction task was management succession events in financial news stories.

- ***MUC-7*** **(1998)**: This is the final conference of the series. Hence it was aiming at giving an overall evaluation of the tasks and results which have been used in previous MUCs. Table 2.3 summarises these changes beginning with MUC-3. In MUC-7, there were more international sites participating than ever before, and

more data was provided for training and dry run and it was maintained through all of the updates to the guidelines during the evaluation cycle (Chinchor, 2001).

Table 2.3: Tasks Evaluated in MUC-3 through MUC-7 (Chinchor, 2001)

Evaluation Tasks	***Named Entity***	***Coreference***	***Template Element***	***Template Relation***	***Scenario Template***	***Multilingual***
MUC-3					YES	
MUC-4					YES	
MUC-5					YES	YES
MUC-6	YES	YES	YES		YES	
MUC-7	YES	YES	YES	YES	YES	

2.2.2 Information Extraction Challenges

Text usually contains a large amount of names such as person names, city names, organization names, etc. in order to extract these names from the text it's impracticable to list all these names in several lists, furthermore these names can appear in the text with more challenges such as:

I. The overlap between the lists of entity:

- Organisation vs. Location:

"England won the World Cup" vs. "The World Cup took place in England".

- Person vs. Artefact:

"The ham sandwich wants his bill." vs. "Bring me a ham sandwich."

- Currency vs. weight:

"Mr. Jones lost 25 pounds…" Did he lose 25 pounds of weight or 25 pounds of British currency?

- Date vs. Time:

"1945" is that time or date?

In the same way, issues of style, structure, domain, genre, punctuation, spelling, spacing, formatting are also problematic as shown in the following:

II. The variation:

There are lots of ways to express the same event such as:

- "The Royal Bank of Scotland plc"
- "The Royal Bank of Scotland"
- "The Royal plc"
- "The Royal"
- "RBS"
- Or John Smith, Mr Smith, John.

III. Overlap between the categories

- "Hope" and "Lost" as proper names (location)
- "Hope" and "Lost" as common nouns

IV. Complex entities with conjunction:

- "China International Trust and Investment Corp"
- "Mason, Daily and Partners"

There also exist more general problems of robustness and portability, e.g. how can a system recognize NEs when they appear in headlines or at the beginning of sentences where capitalization information is missing? Some of these challenges exist in the Arabic language and these will be highlighted in chapter 3. The experience of the MUCs has demonstrated that IE is a difficult task (Appelt and Israel, 1999).

2.2.3 Information Extraction Tasks

The IE term is an umbrella that covers a number of tasks which have been identified by MUCs. There were four evaluated tasks; Named Entity recognition, Coreference resolution, Template element filling and Scenario template filling. IE tasks needs to be utilized over the text as shown in Figure 2.1.

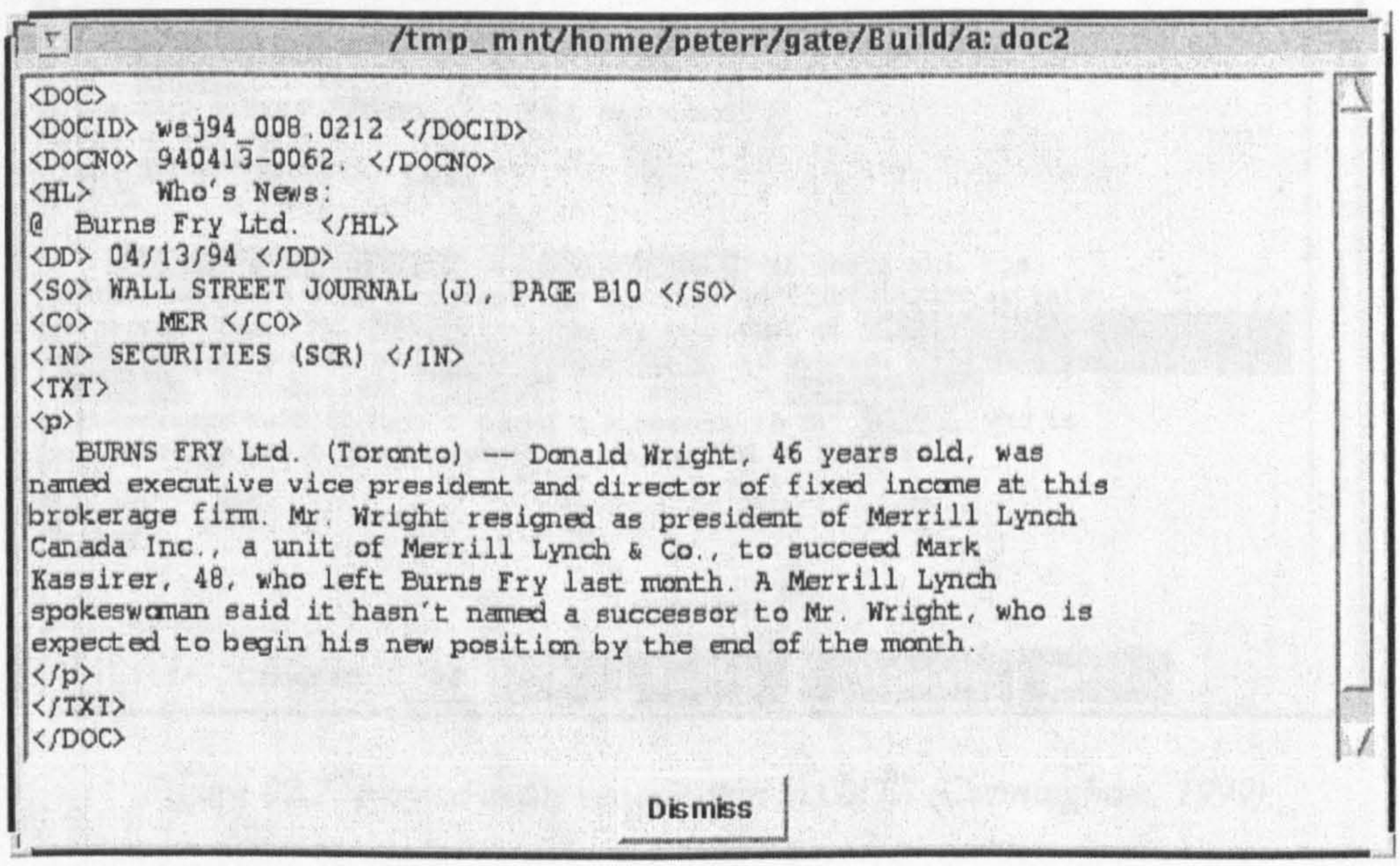

Figure 20.1: An Example of a Text (Cunningham, 1999)

2.2.3.1 Named Entity Recognition

A Named Entity (NE), which was first defined in MUC 6, is the recognition and classification of defined named entities such as organizations (companies, government organisations, committees, etc), persons, locations (cities, countries, rivers, etc) dates and time expressions and monetary amounts (percent, money, weight etc) (Gaizauskas, 1998). An example of a named entity annotation is shown in Figure 2.2. NE extraction is an essential tool for term extraction that is important in various NLP applications. For

instance, automatic text summarization systems can be improved by using NE, as they provide important cues for identifying relevant segments in texts. Other uses of NE taggers are accurate internet search engines, IE, automatic speech recognition, question answering and MT.

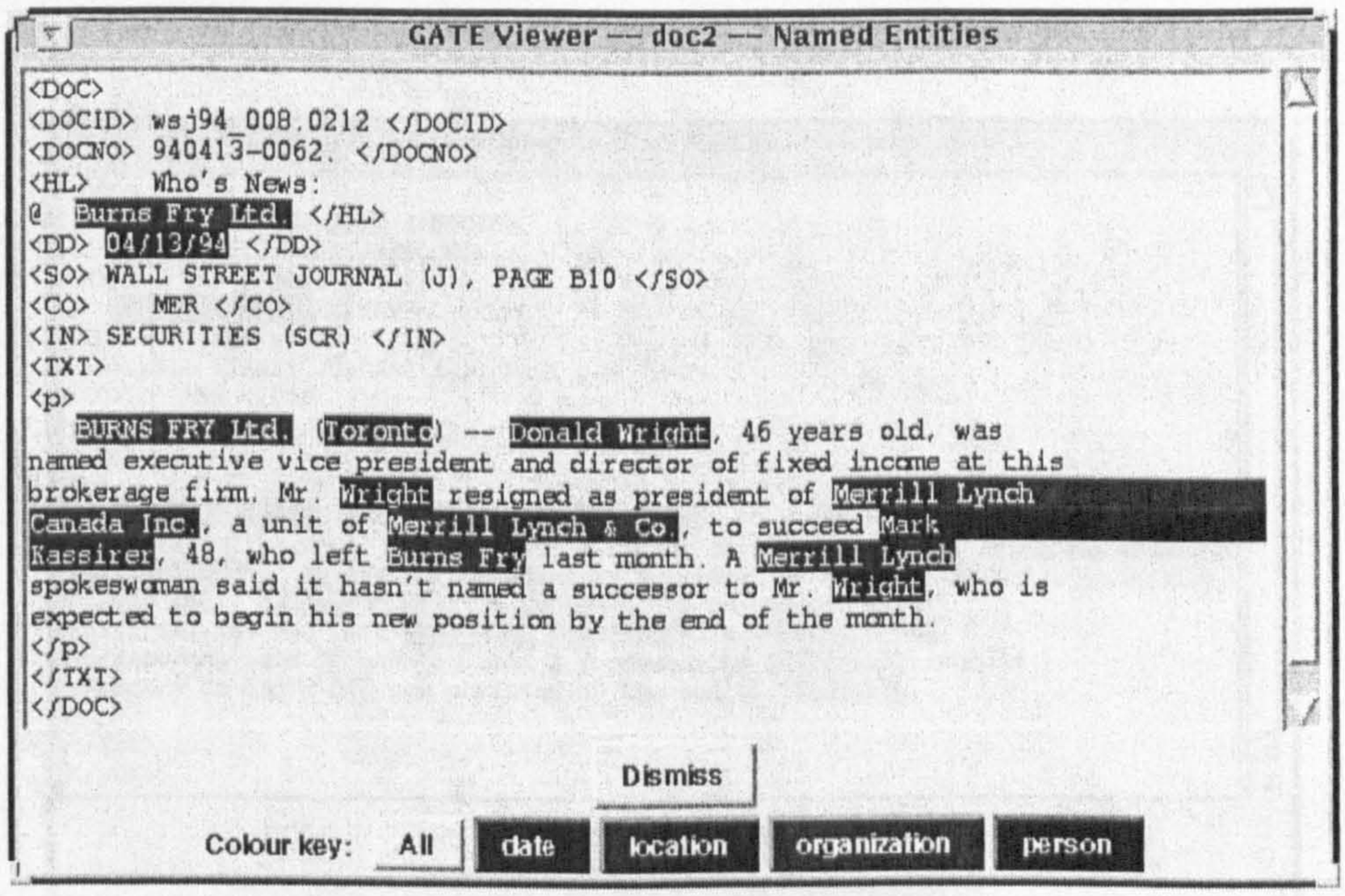

Figure 02.2: Named entity recognition in GATE (Cunningham, 1999)

Correct recognition of NEs is a significant issue for MT research and for the development of MT systems. Primary, translation of proper names often needs different approaches and techniques than the translation of other types of words (Babych, 2003), and proper nouns can be translated with higher accuracy using NE taggers. NE can be valuable in several NLP applications. However, names represent a large percentage of unknown words in a text. Moreover, names are considered as a crucial source of information in a text when extracting contents, clarifying a subject or identifying related documents in IR systems (Rau, 1991). Therefore, the accuracy of tools such as chunkers and parsers in IE systems rely on the recognition of these names.

2.2.3.2 Coreference Resolution

This task requires the identification of expressions in the text that refer to the same object, set or activity as shown in Figure 2.3. The coreference relation will be marked between elements of the following categories: nouns, noun phrases and pronouns (Gaizauskas, 1998).

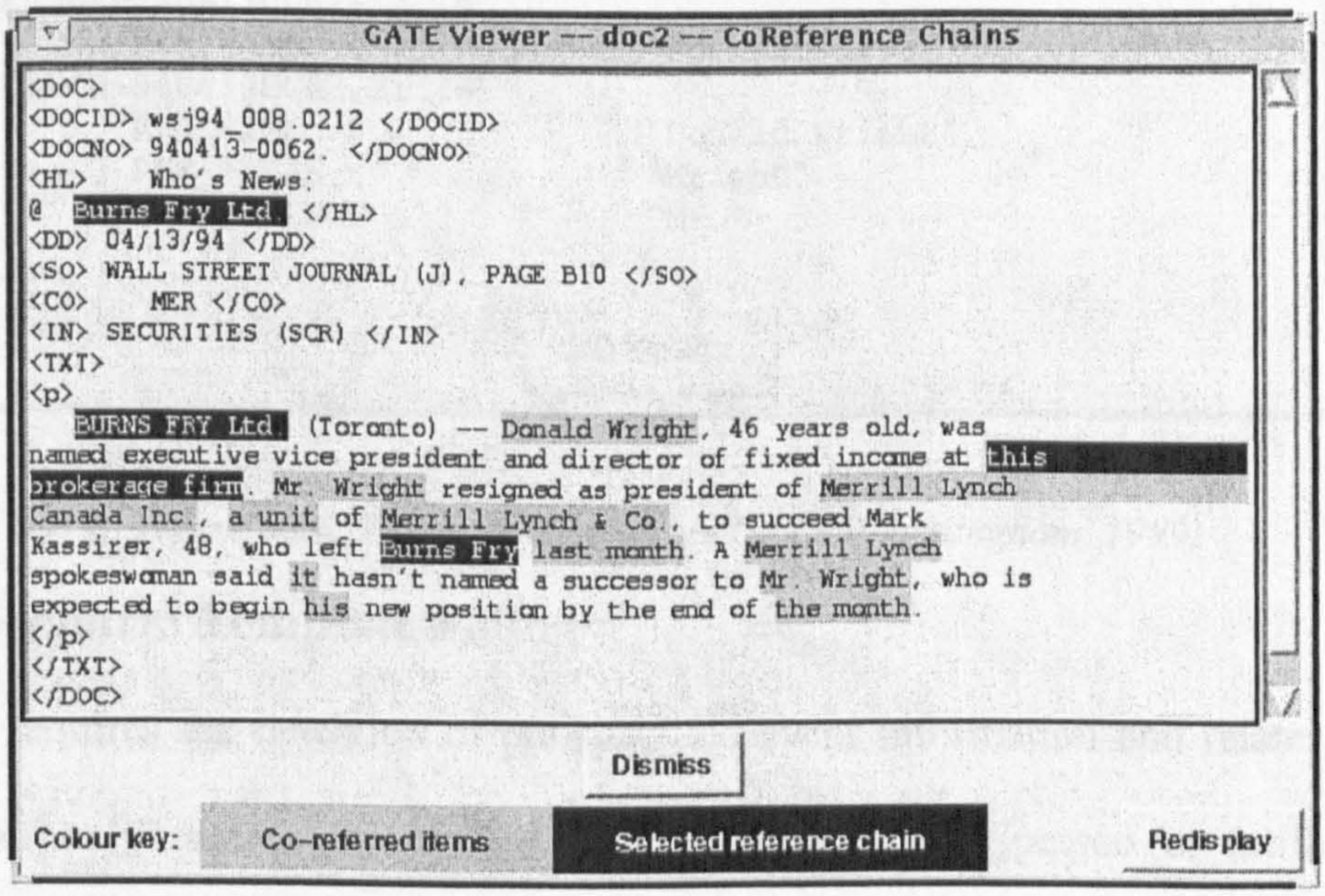

Figure 02.3: Coreference resolution in GATE (Cunningham, 1999)

2.2.3.3 Template Element Filling

This task requires the filling of small scale templates wherever they occurred in the text using the basic information related to organization, person and artifact entities, as shown in Figure 2.4.

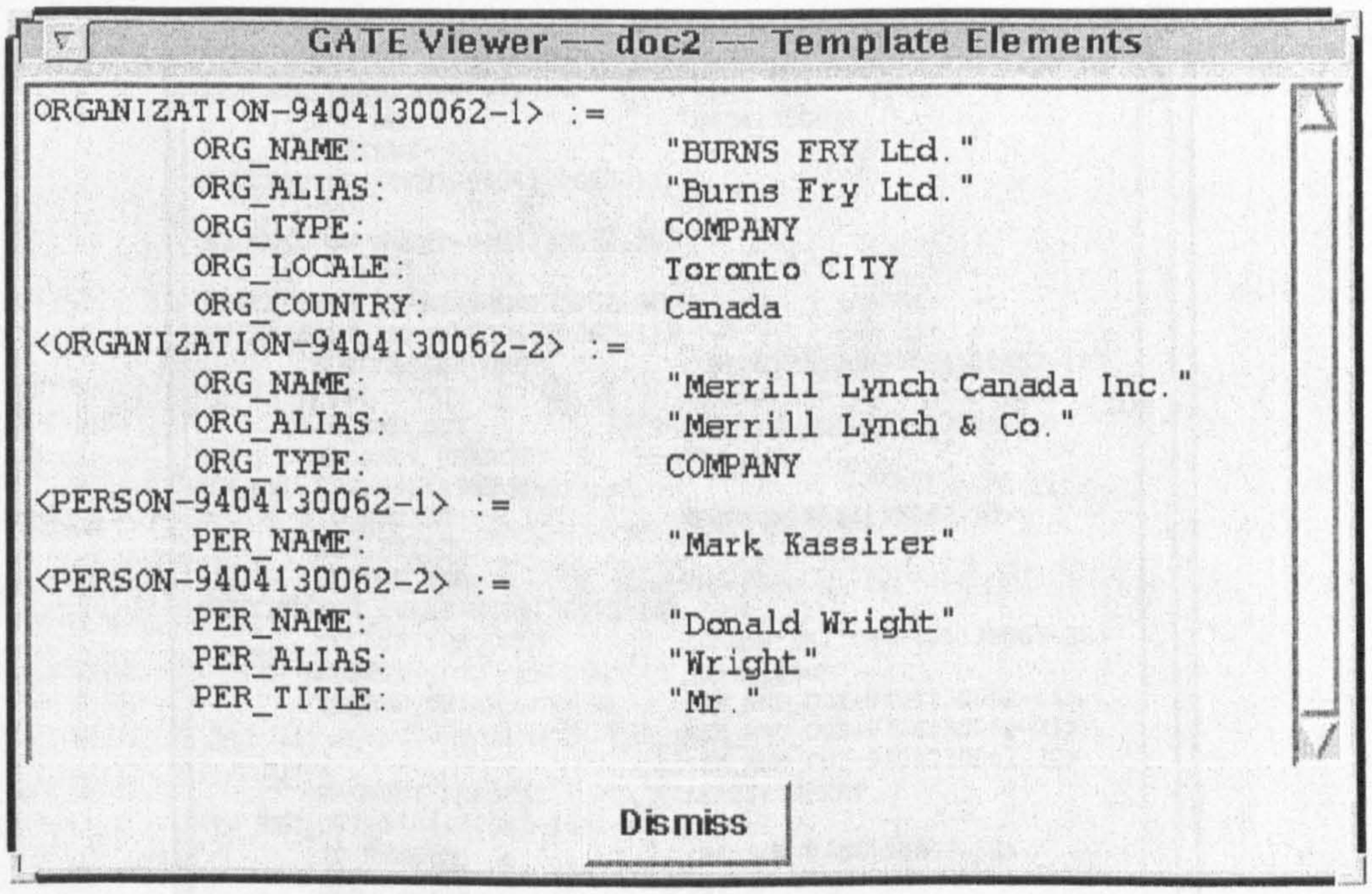

Figure 02.4: Template elements in GATE (Cunningham, 1999)

2.2.3.4 Scenario Template Filling

This task requires the detection of pre-specified event information and relates the event information to particular information such as organization, person or artifact entities involved in the event, as shown in Figure 2.5. The filling of the slots in the scenario template was generally a difficult task for systems.

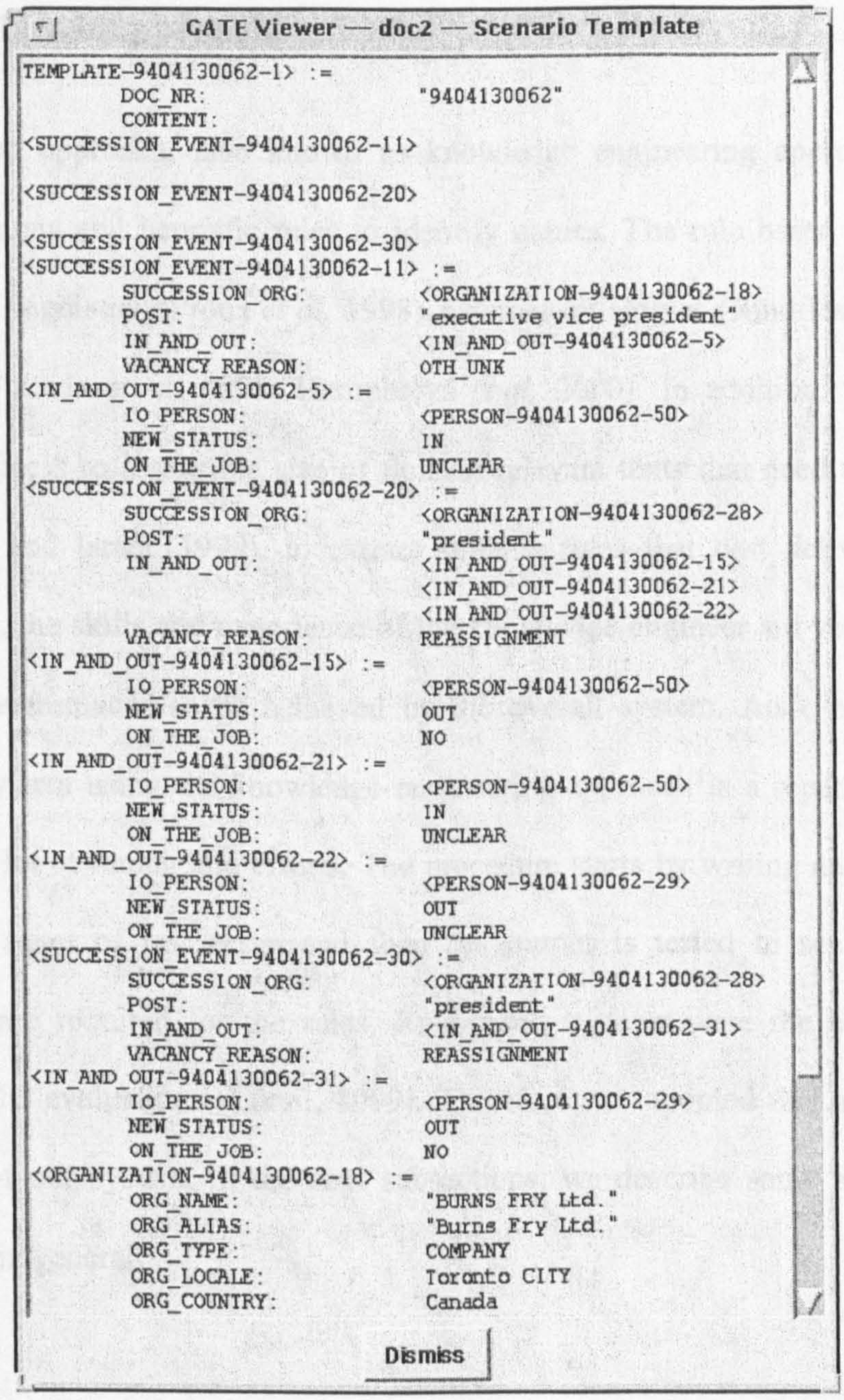

Figure 02.5: Scenario template in GATE (Cunningham, 1999)

2.2.4 Information Extraction Approaches

There are two basic approaches to IE system development: the Rule based (Knowledge Engineering) Approach and the Automatic Training Approach (statistical) (Appelt and Israel, 1999). In the following subsections, we describe each of these two approaches.

2.2.4.1 Rule based approach

The Rule based approach, also known as knowledge engineering approach, relies on regular expressions and heuristic rules to identify names. The rule based approach also relies more on linguistics (Proux *et al.* 1998), external ontologies (Rindflesch *et al.* 2000) and context (Fukuda *et al.* 1998; Humphreys *et al.* 2000). In addition, the knowledge engineer has access to a sensible size of domain-relevant texts that need to be manually tested (Appelt and Israel, 1999), to extract suitable rules that best deliver the needed output. Clearly, the skills and experience of the knowledge engineer are very important to improve the performance to be achieved by the overall system. Also, building a high performance system using the knowledge engineering approach is a repetitive procedure that requires a lot of tuning and efforts. The procedure starts by writing and running a set of rules over a set of test texts, and then the output is tested to see what suitable modifications are required for the rules. Rule-based systems were the best performing systems at MUC evaluations (Eikvil, 1999). Therefore, we adopted this approach in the development of our system. In the next subsections, we describe some aspects of rule-based systems in general.

2.2.4.2 Rule Based system

Rules are the basic components of a rule based system. Knowledge, usually extracted from experts, is encoded as a set of rules. Rules are like the if-then statements in traditional programming languages (Friedman-Hill, 2003). For instance, a statement like "A present verb in a third person ends with an s" might be written as a rule like:

IF w is a verb AND w is in present tense AND w is in third person THEN w ends with s

The IF part of a rule written in this form is often called left hand side (LHS), premises, and the then part is the right hand side (RHS), conclusions. A rule-based system is a system that uses rules to derive conclusions from premises (Friedman-Hill, 2003). Rule based systems are declarative and include only the important details of a solution. Declarative programming is often seen as the natural way to tackle problems involving control, diagnosis, prediction, classification, pattern recognition; in short, many problems without clear algorithmic solutions (Friedman-Hill, 2003). Rule-based programs are simple and are made up of discrete rules that are applied to some subset of the problem. The rule base also referred to as knowledge base stores all relevant information such as rules, data and relationships between data. Once the rules are defined, we usually develop a rule engines also referred to as inference engine which applies the IF-THEN rules to some defined data and takes appropriate actions. The main purpose of the inference engine is to seek information and relationships from the knowledge base and to provide answers, predictions, and suggestions in the way a human expert would (Moisiadis et al, 2008). Forward and backward chaining are two known inference methods used derive facts from rules and data (Abraham, 2005).

However our application contains very limit rules and we structured them as a separate package. The rules do not change so often and no inference engine has been developed for our system as the rules are very simple and directly coded in Java. The development of our rules will be described in details in chapter 5.

2.2.4.3 Overview of Systems Adopted Rule Based Approach

Earlier IE systems were mainly based on pattern-matching grammars. Hobbs et al. (1996) proposed the FASTUS system, an IE system based on finite state automata. FASTUS searches the input text for trigger words and uses the rules attached with them to extract the required data. It achieved 44% recall and 55% precision on an IE task from 100 texts, the state-of-the-art performance in 1993. In the New York University's Proteus IE system (Grishman, 1997), names were identified by a set of patterns (regular expressions) that were represented in terms of part-of-speech, syntactic structures, orthographic features like capitalization and a dictionary of name list. Figure 2.6 illustrates the architecture of the Proteus system. Other rule-based IE systems include AutoSlog (Riloff, 1996), and RAPIER (Califf and Mooney 1997).

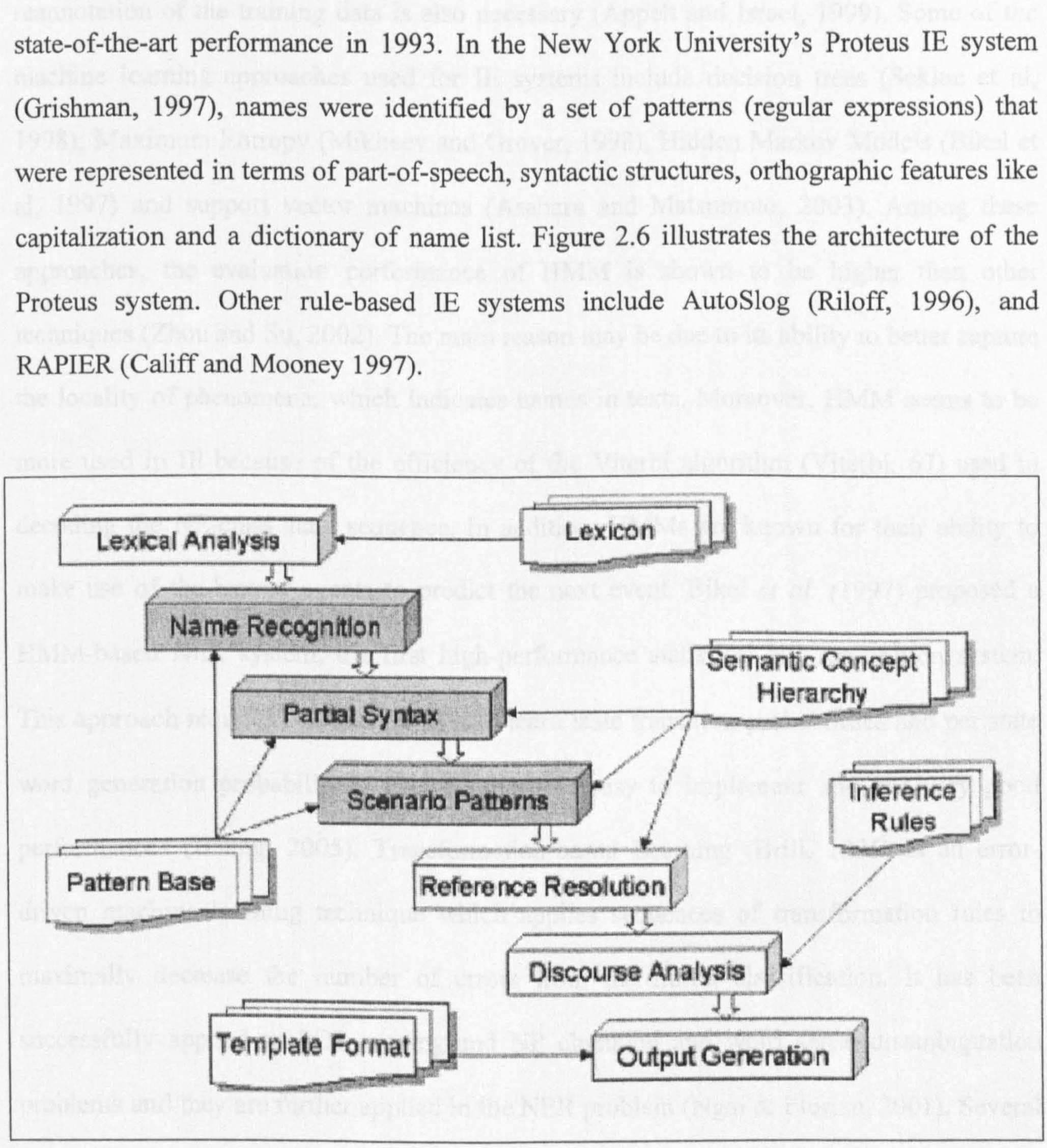

Figure 02.6: Proteus system architecture. (Kaiser and Miksch, 2005)

2.2.4.4 Automatic Training Approach

This approach requires someone who has enough knowledge about the domain and the tasks of the system to annotate the texts appropriately. However, statistical parsing needs a large amount of training data and labelling parse trees manually from text. Therefore it's an expensive process. In case any change occurs to the IE system specifications, reannotation of the training data is also necessary (Appelt and Israel, 1999). Some of the machine learning approaches used for IE systems include decision trees (Sekine et al, 1998); Maximum Entropy (Mikheev and Grover, 1998), Hidden Markov Models (Bikel et al, 1997) and support vector machines (Asahara and Matsumoto, 2003). Among these approaches, the evaluation performance of HMM is shown to be higher than other techniques (Zhou and Su, 2002). The main reason may be due to its ability to better capture the locality of phenomena, which indicates names in texts. Moreover, HMM seems to be more used in IE because of the efficiency of the Viterbi algorithm (Viterbi, 67) used in decoding the NE-class state sequence. In addition, HMMs are known for their ability to make use of the history events to predict the next event. Bikel *et al.* (1997) proposed a HMM-based NER system, the first high-performance statistical NE recognition system. This approach requires NE labeled data to learn state transition probabilities and per state word generation probabilities. This approach is easy to implement and has very good performances (Huang, 2005). Transformation-based Learning (Brill, 1995) is an error-driven machine learning technique which applies sequences of transformation rules to maximally decrease the number of errors from the initial classification. It has been successfully applied to POS tagging and NP chunking and word sense disambiguation problems and they are further applied in the NER problem (Ngai & Florian, 2001). Several statistical systems are developed for the English language such as IdentiFinder (Bikel et al, 1999) which architecture is illustrated in Figure 2.7, IdentiFinder is a HMM that learns to

recognize and classify names, dates, times and numerical quantities, The Kent Ridge Digital Labs system (Bai and Wu, 1998), MENE (Borthwick et al, 1998), and RoboTag (Bennett et al, 1997), etc.

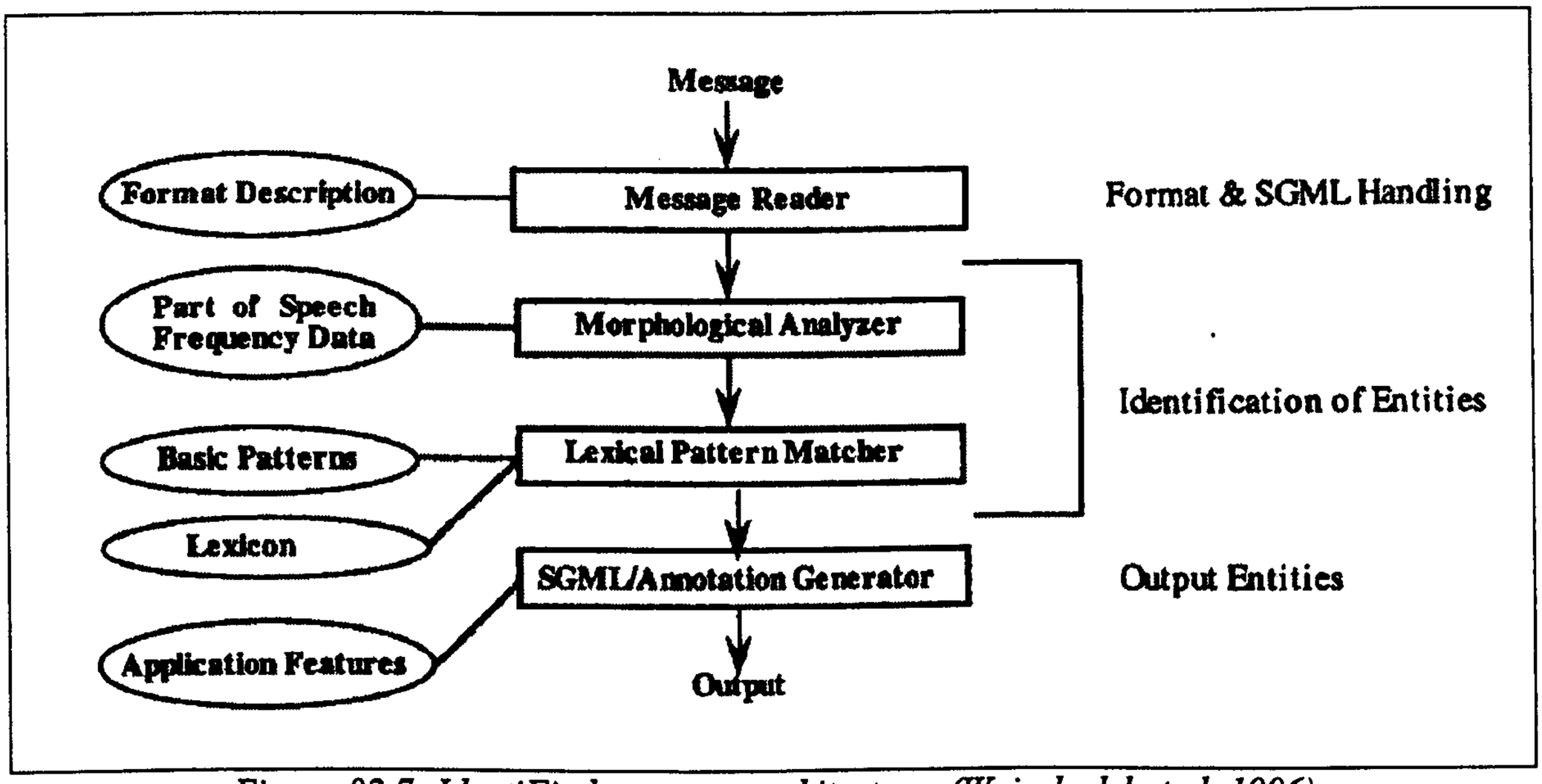

Figure 02.7: IdentiFinder system architecture. (Weischedel et al. 1996)

2.2.5 Information Extraction Architecture

In summary IE systems are composed of four main modules which are tokenization, morphological and lexical processing, syntactic analysis, and Domain analysis (Kaiser and Miksch, 2005) as shown in the left-hand column of Figure 2.8. However, depending on what one is concerned with, some phases may not be essential. Furthermore, to the components in the left-hand column, IE systems may comprise components from the right-hand column, depending on the particular requirements of the application. However, Appelt and Israel (1999) summarize the factors which will affect the system whether it requires additional components over the four essential components as follows:

- Language of the text. Some languages like Semitic languages will require morphological and word segmentation processing.

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

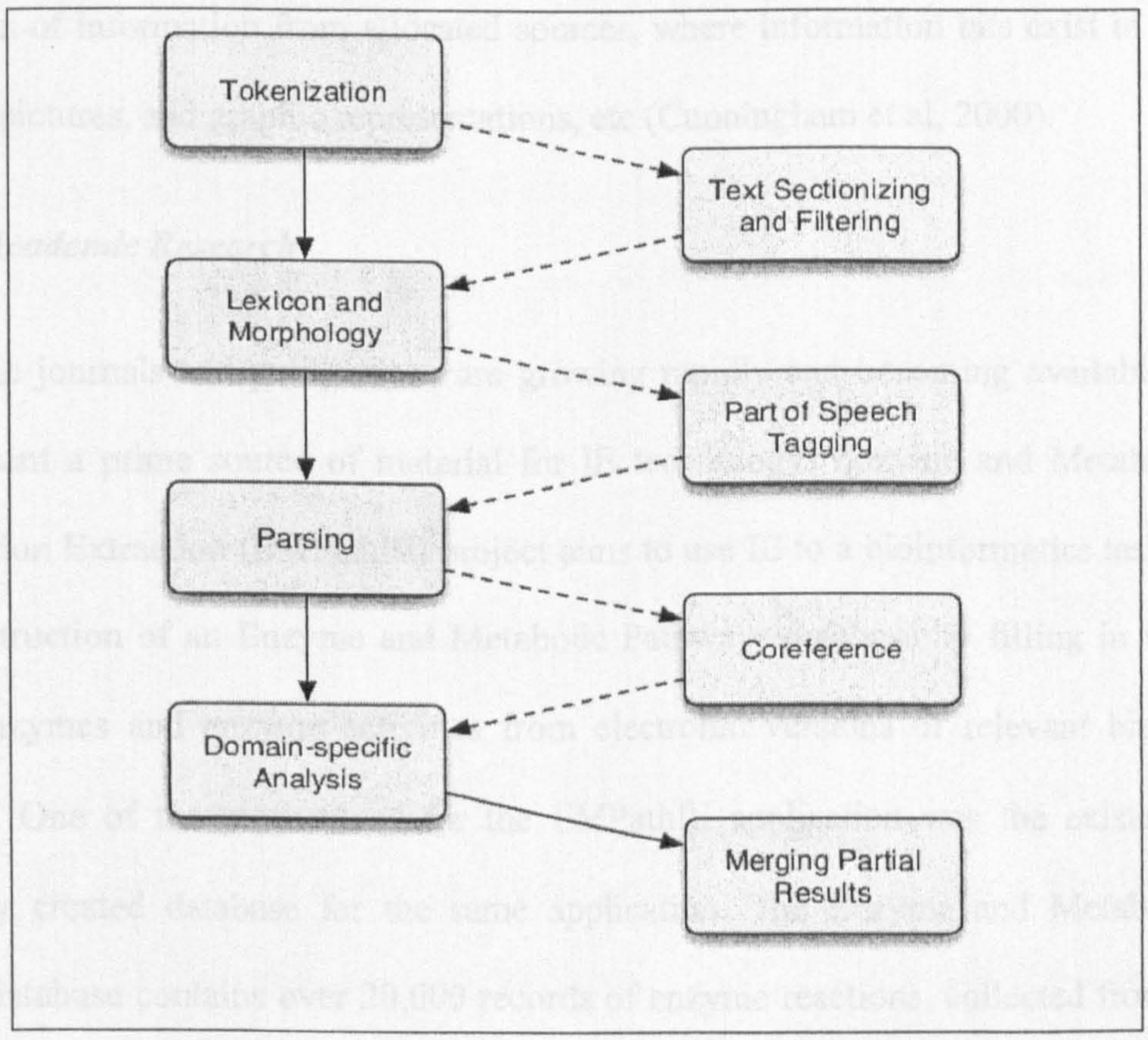

Figure 02.8: The architecture of information extraction systems (Kaiser and Miksch, 2005)

2.2.6 Information Extraction Applications

There are a number of different applications where information extraction can have a practical use:

i. Police

The AVENTINUS Project aims at supporting an Advanced Information System for Multinational Drug Enforcement (Kokkinakis, 1998). It is an EU funded research and development programme set up to build an information system for multinational drug enforcement. The goal of the project is to support drug enforcement with multilingual linguistic expertise; hence the project is working to build tools to assist police in criminal investigations relating to drug trafficking. The project includes multilingual access and extraction of information from allocated sources, where information bits exist in the form of texts, pictures, and graphic representations, etc (Cunningham et al, 2000).

ii. Academic Research

Academic journals and publications are growing rapidly and becoming available on-line and present a prime source of material for IE technology. Enzyme and Metabolic Path Information Extraction (EMPathIE) project aims to use IE to a bioinformatics task such as the construction of an Enzyme and Metabolic Pathways database by filling in templates about enzymes and enzyme activities from electronic versions of relevant bimolecular journals. One of the motivations for the EMPathIE application was the existence of a manually created database for the same application. The Enzyme and Metabolic Path (EMP) database contains over 20,000 records of enzyme reactions, collected from journal articles published since 1964 (Selkov et al., 1996). Figure 2.9 illustrates the modules of the EMPathIE system.

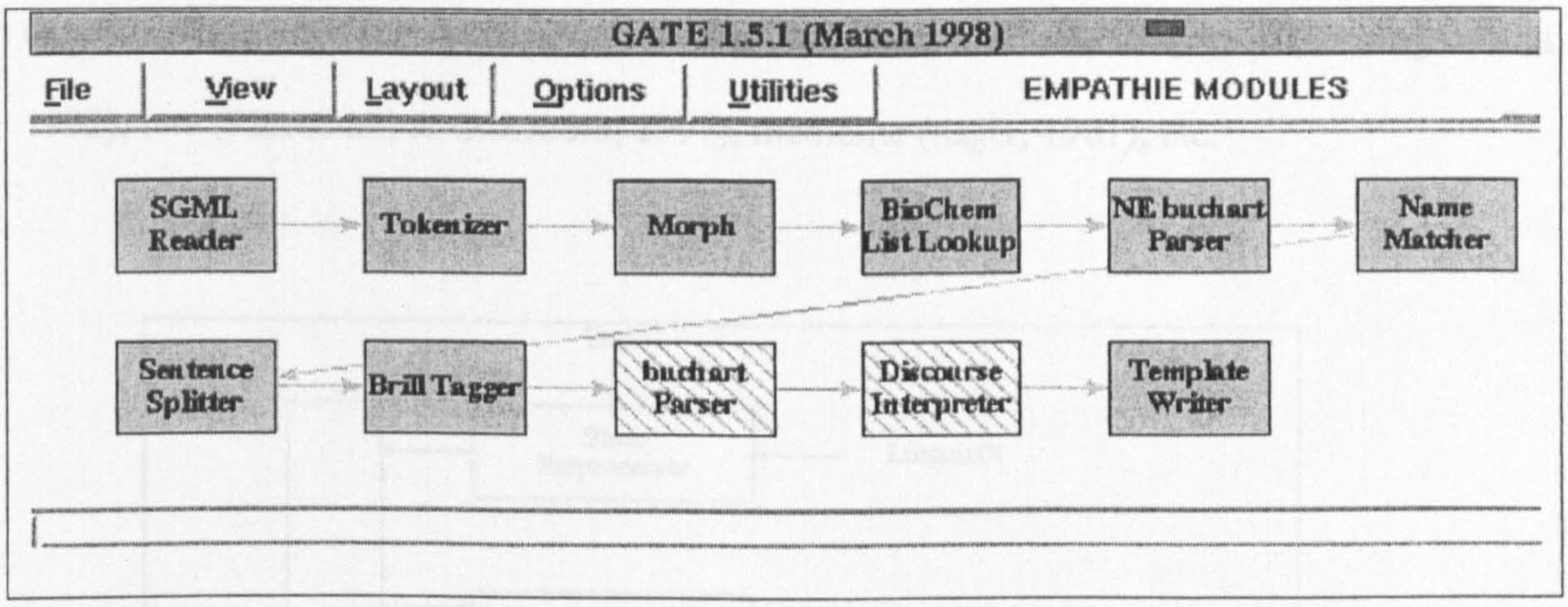

Figure 2.9: EMPathIE system modules within GATE (Humphreys, 2000)

iii. Employment

The TRans European Employment (TREE) aims to provide a system where employees would be able to read job opportunity announcements in any of the several European languages. It summarizes the extracted adverts into a base schema. TREE therefore offers two significant services: intelligent search and summarization in one hand, and the production of outputs in the desired language in the other hand (Somers *et al.*, 1997).

Finance

The Fast Accurate Categorization of Information using Language Engineering (FACILE) project (Black et al, 1998) which uses IE techniques, aims to achieve the filtering of news by fine-grained knowledge-based categorization, Figure 2.10 illustrates the architecture of the system. The system has been initially developed as an applied research project involving research centers, industrial bodies and end-user organizations. The project is a considerable success story in the financial field. Moreover, the system was adopted by the main Italian financial news agency and has been running continuously since January 1998 (Ciravegna, 2000). The FACILE system categorizes texts in four languages: English, German, Italian and Spanish. However, these are just application domains and there are

several other application areas were IE is effective such as fault diagnosis (Ciravegna et al, 1992), law (Piltrosanti & Graziadio, 1997), medicine (Sager, 1981), etc.

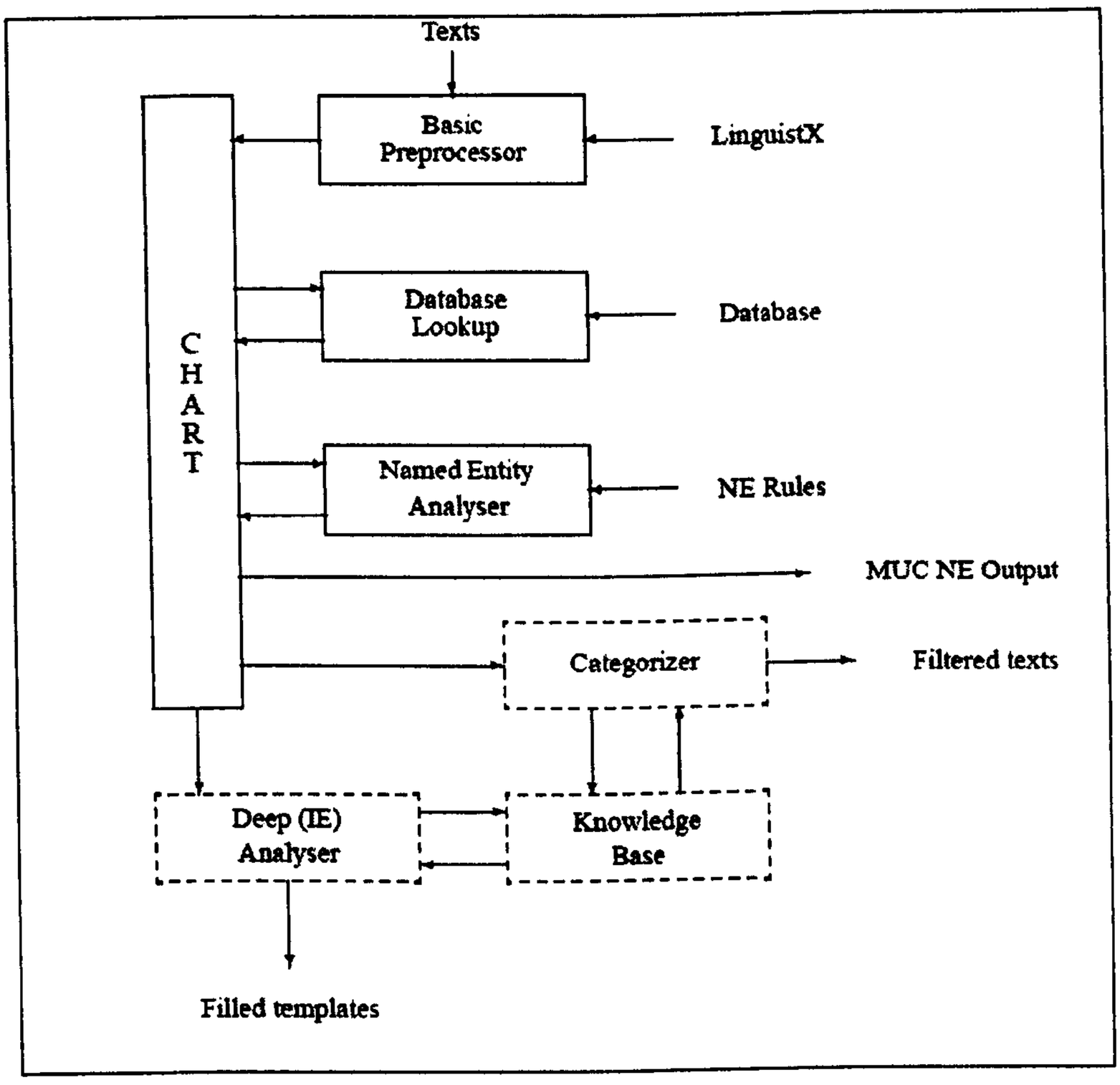

Figure 2.10: FACILE system architecture (Black, 1998)

2.3 Arabic Natural Language Processing Challenges

Arabic NLP is still in its initial stages compared to the work on other languages such as English, which has benefited from the extensive research in this field. There are some aspects that slow down progress in Arabic NLP compared to the accomplishments in other European languages (Al-Daimi and Abdel-Amir, 1994). Semitic languages in general and Arabic scripted languages in particular, present a challenge to the automated approaches for Proper Names and/or NE recognition. That is why systems dealing with Semitic or

Arabic NE have to adopt different techniques to overcome such challenges which can be summarised as follows:

- Arabic is a highly inflectional and derivational language, from one root several words can be obtained, as shown in Table 2.4. This issue makes the morphological analysis a very complex task. Its grammatical system is traditionally described in terms of a root-and-pattern structure, with about 10,000 roots (Ali, 1988).

Table 02.4: Word derivation from the root "ktb"

Word	***The translation***
كتب	Write
كتاب	Book
كاتب	Writer
مكتوب	Written
كتيب	Small book

- The absence of diacritics (which represent most vowels) in the written texts creates ambiguity and therefore, complex morphological rules are required to identify the tokens and parse the text. For example, the word (على) without vowels can mean the proper name (Ali) or the preposition (on). This ambiguity will be a critical problem in language processing as an Arabic word can have several meanings.
- Several types of affix are agglutinated to the beginning and the end of the words: prefixes, suffixes and infixes.
- The writing direction is from right-to-left and some of the characters change their shapes based on their location in the word.
- Capitalization is not used in Arabic, which makes it hard to identify proper names, acronyms, and abbreviations.
- Words are separated by space and other punctuation marks. Nevertheless, prepositions are agglutinated to the word appearing after them, making the boundary between the word and the preposition invisible.

- Most of the research in Arabic NLP systems mainly concentrated on the field of morphological analysis (Ditters, 2001 and Jaccarini, 2001).

In addition to the above linguistic issues, there is also a lack of Arabic corpora, lexicons, and machine readable dictionaries, which are essential to advance research in different areas (Samy, 2005). Obviously, Arabic is also very different from European languages at syntactic and semantic levels.

2.4 Related Work

In this section we review and evaluate some Named Entity recognition systems developed for the Arabic language and we discuss some of the issues related to these systems in particular and to the future of NE recognition in Arabic in general. However it is worth noting that these systems adopted various approaches and used different corpora. On the other hand there are some systems such as (Harmain and Aljohar, 2001) designed as a general framework for an Arabic IE system but without any implementation of the suggested system components. Work on Arabic language can be classified into two categories, those systems that work on transliterated Arabic texts and those working on pure Arabic texts. In this chapter we will give a general view of these systems based on the information extraction method used (rule-based or statistical) and whether the system is dealing with raw Arabic texts or transliterated texts.

2.4.1 Systems Using Transliterated Arabic Texts

Transliteration is the process of representing words from one language using the alphabet or writing system of another language (Arbabi et al., 1994). Transliteration from a language like Arabic would differ depending on the target language. An example is the

Arabic name مصطفى, which could be transliterated into English as 'Mustafa' or 'Mustapha', while a likely French transliteration would be 'Moustafa' or 'Moustapha'. As multilingual NER is concerned, the transliteration of the NE included alternative spelling variants where the original language of the name is usually not known. Several variants could also be found in the same language.

2.4.1.1 Rule Based Technique

Samy et al. (2005) adopted an approach that relies on two main types of resources; parallel corpora and previously developed tools for other languages. Their implementation relies on this basic assumption: "Given a pair of sentences where each is the translation of the other; and given that in one sentence one or more NE were detected, then the corresponding aligned sentence should contain the same NE either translated or transliterated". While they report high precision and recall, it should be noted that their approach is applicable only when a parallel corpus is available.

2.4.1.2 Statistical Techniques

Larkey *et al.* (2003) have presented a statistical technique for English to Arabic transliteration. This technique requires no heuristics or linguistic knowledge of either language. This technique learns translation probabilities between English and Arabic characters from a training sample of pairs of transliterated words from the two languages. The data they used to train and test the system was a parallel list of 148,599 English and Arabic proper nouns, the list contained both person names and place names. They showed the importance of the proper names component in cross language tasks involving searching, tracking, retrieving, or extracting information. In particular, they concluded that

a combination of static proper name (English-Arabic) translation, plus transliteration provides a successful solution. Zitouni *et al*, (2005) have adopted a statistical approach for the entity detection and recognition (EDR). In their work, a *mention* can be either named (e.g. John Mayor), nominal (the president) or pronominal (she, it). An entity is the aggregate of all the mentions (of any level) which refer to one conceptual entity. This extended definition of the entity has proved the suitability of their approach. Benajiba and Rosso (2008) produced a Named Entity Recognition (NER) system using Support Vector Machines (SVMs) and a combination of both language independent and language dependent features for Arabic NER. However, using SVM for the Arabic language is inappropriate for various reasons. First the input to the SVM tagger has to be a transliterated (written in Latin letters) Arabic text (Shamsi and Guessoum, 2006). Moreover the SVM model is weak in estimating a classification for very low frequency words; also it was observed that training the SVM model is slow (Takeuchi and Collier, 2002). Second, in their system they employ sets of features. In addition they use the nationality as contextual and a lexical feature, they argue that marking nationalities in the input text is useful information in detecting NEs as it used as precursors to recognizing NE. However they illustrated this issue by giving the example shown in Table 2.5.

Table 2.5: An example showing a proper noun after a keyword (Benajiba and Rosso, 2008)

The Arabic Sentence	***The English translation***
وصرح الرئيس الإيراني محمود	And the Iranian president Mahmoud declared

But, as we illustrated in the examples given in Figure 7.6 the Arabic phrase can be established in different forms in the text. Hence we cannot constantly mark the word next to the nationality as a NE. Moreover the nationality sometimes does not refer to the human at all as the examples given in Table 2.6. In addition they use a feature called

Corresponding English Capitalization (CAP). The insight is that if the translation begins with a capital letter, then it is most probably a NE. However this assumption is imprecise, and to clarify this point consider this examples given in Table 2.6 which are not Named Entities.

Table 0.6: Non named entity words starting with a capital letter.

The Arabic Sentence	*The English translation*
الدجاج الفرنسي	The French chickens
الأحذية الإيطالية	The Italian shoes
البقر البريطاني	The British cows
السيجارة الأمريكية	The American cigarette
الأرز الصيني	The Chinese rice
النفط السعودي	The Saudi oil

2.4.1.3 Dictionary Technique

Pouliquen *et al.* (2005) developed a tool for multilingual person name recognition that focuses on the "Who" part of the analysis of large news text. However, they consider in their system, as do most of the approaches used in the field, the proper noun next to the trigger word as shown in figure 2.11 where the words in italics show the trigger words (former Prime Minister,رئيس الوزراء السابق) and (Rafik Hariri,رفيق الحريري) the proper noun.

Language	*text*
English	…death of *former Prime Minister* **Rafik Hariri**, blamed by many opposition…
Spanish	…asesinato del *exprimer ministro* **Rafic al-Hariri**, que la oposición atribuyó…
French	…l'assassinat de l'*ex-dirigeant* **Rafic Hariri** et le départ du chef de la …
Dutch	na de moord op *oud-premier* **Rafiq al-Hariri** gingen gisteren bijna een…
German	… *libanesischen Regierungschef* **Rafik Hariri** vor einem Monat wichtige…
Slovene	danjega *libanonskega premiera* **Rafika Harirlja**. Libanonska opozicija si…
Estonian	mööduniselе *ekspeaminister* **Rafik al-Hariri** surma põhjustanud…
Arabic	…اغتيال *رئيس الوزراء السابق* **رفيق الحريري** بأيادٍ يهودية وما حدث سابقا …
Russian	…*Бывший премьер-министр Ливана* **Рафик Харири**, который…

Figure 2.11 : Examples of a proper noun next to a trigger word (Pouliquen et al., 2005)

Kashani et al. (2006) proposes a novel spelling based method for the automatic transliteration of proper nouns from Arabic to English which exploits various types of letter-based alignments. The approach consists of three phases: the first phase uses single letter alignments, the second phase uses alignments over groups of letters to deal with diacritics and missing vowels in the English output, and the third phase exploits various knowledge sources to repair any remaining errors. They used a list of name pairs, i.e. names written in Arabic and correctly transliterated into English. They used two different sources. The first source was a set of named entities and their transliterations which are annotated in the LDC Arabic Treebank. The second source was the Arabic-English Parallel News corpus automatically tagged using an entity tagger. They aligned the tagged parallel texts and extracted the aligned names. In total, 9660 name pairs are prepared from these two sources.

2.4.2 Review of Work Using Raw Arabic Language

2.4.2.1 Rule Based Technique

Saleem (2004) presented a technique to extract names from text by building a database and graphs to represent the words that might form a name and the relationships between them. First he marks the phrases that might include names, second he build graphs to represent the words in these phrases and the relationships between them, third he apply rules to find the names. He tested the technique on 500 articles from the *Al-Raya* newspaper *(2003),* published in Qatar and the system extracted 78.4% of the names found in the text. As we illustrated above the main elementary tools for processing any language for any NLP application are the morphological analyzer and POS tagger. In general, the morphological analyzer gives multiple (ambiguous) labels or tags for a word and the POS tagger is then

used to resolve the ambiguity by assigning the most appropriate tag. Whenever the results of the POS tagger are precise then the entire results of the application are accurate and vice versa. However the majority of the systems used in the field adopted approach that combined several rules and predefined name lists (gazetteers). Shaalan and Raza (2007) developed a system, Person Name Entity Recognition for Arabic (PERA), using a rule-based approach. The system consists of a lexicon, in the form of gazetteer name lists, and a grammar, in the form of regular expressions, which are responsible for recognizing person name entities. The PERA architecture is shown in Figure 2.12.

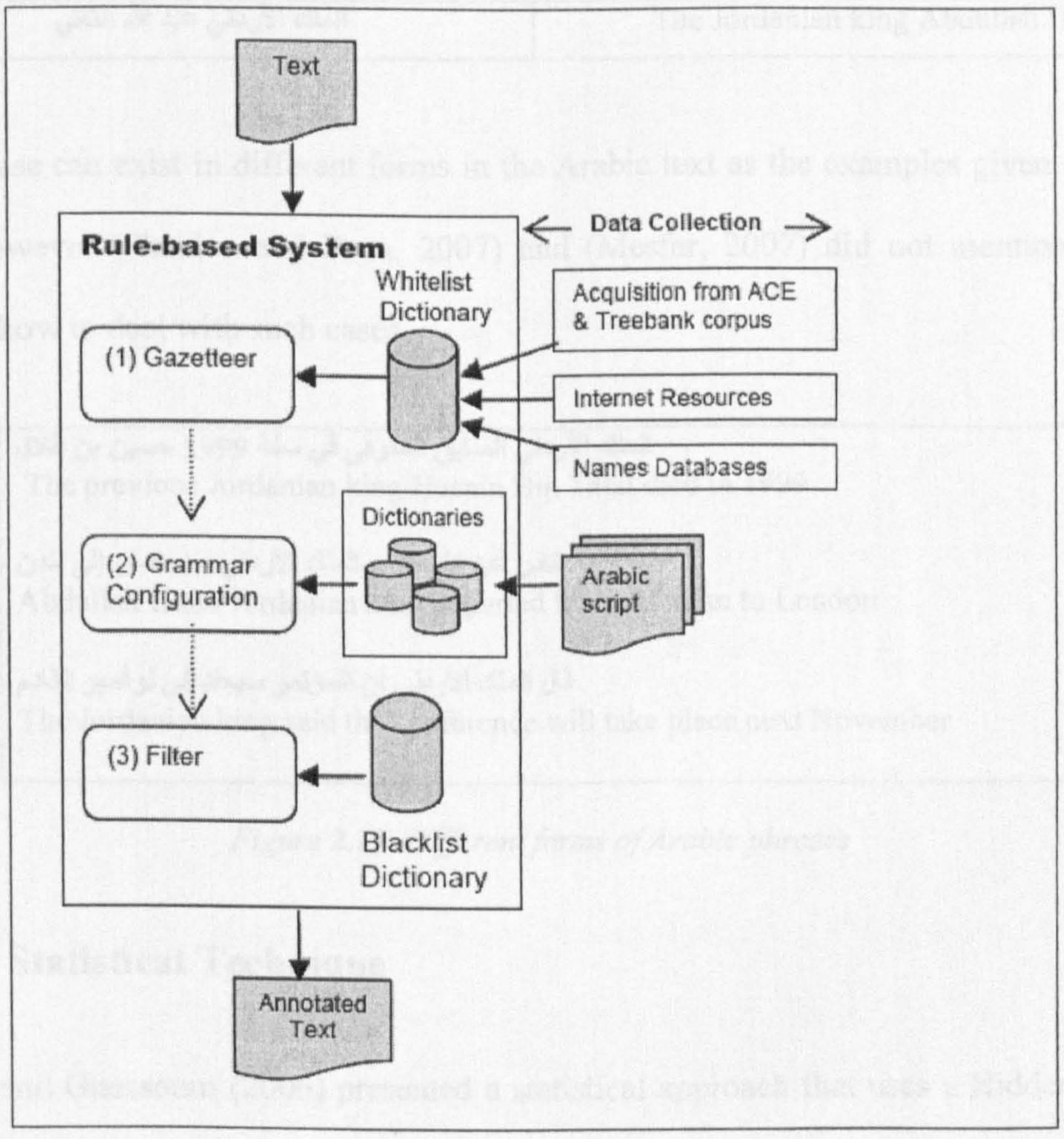

Figure 2.12: The architecture of the PERA System (Shaalan and Raza, 2007)

However the arrangement of the Arabic phrase does not always take one structure. Sometimes the proper noun in the phrase appears next to the keyword and sometimes appears after four or five words after the keyword and sometimes the proper noun appears before the keyword and sometimes the proper noun is completely omitted from the phrase. Consequently we cannot constantly mark the words next to the keyword as a proper noun, as the example in Table 2.7 given by Shaalan and Raza (2007).

Table 02.7: An example of an Arabic phrase

The Arabic phrase	***The English translation***
الملك الأردني عبد الله الثاني	The Jordanian king Abdullah II,

This phrase can exist in different forms in the Arabic text as the examples given in Figure 2.13. However (Shaalan and Raza, 2007) and (Mesfar, 2007) did not mention in their systems how to deal with such cases.

الملك الأردني السابق المتوفى في سنة 1999 حسين بن طلال
The previous Jordanian king Husain Bin Talal died in 1999

غادر عبد الله الثاني الملك الأردني من عمان إلى لندن
Abdullah II the Jordanian king departed from Amman to London

قال الملك الأردني إن المؤتمر سيعقد في نوفمبر القادم
The Jordanian king said the conference will take place next November

Figure 2.13: Different forms of Arabic phrases

2.4.2.2 Statistical Technique

Shamsi and Guessoum (2006) presented a statistical approach that uses a Hidden Markov Model (HMM) to build POS tagging of Arabic text as shows in Figure 2.14. They have developed their corpus of native Arabic articles, which they have manually tagged. The tagger was trained on 27594 nouns, 23554 verbs, 5722 adjectives and 5384 proper nouns.

Their system relies on the output of the Buckwalter morphological analyzer, however the inadequacy of their system is that they select the first output of the Buckwalter as the correct result for a given word. (person name), but the first output of the Buckwalter as shown in figure 2.15 is imperfect verb which means "he made a peace".

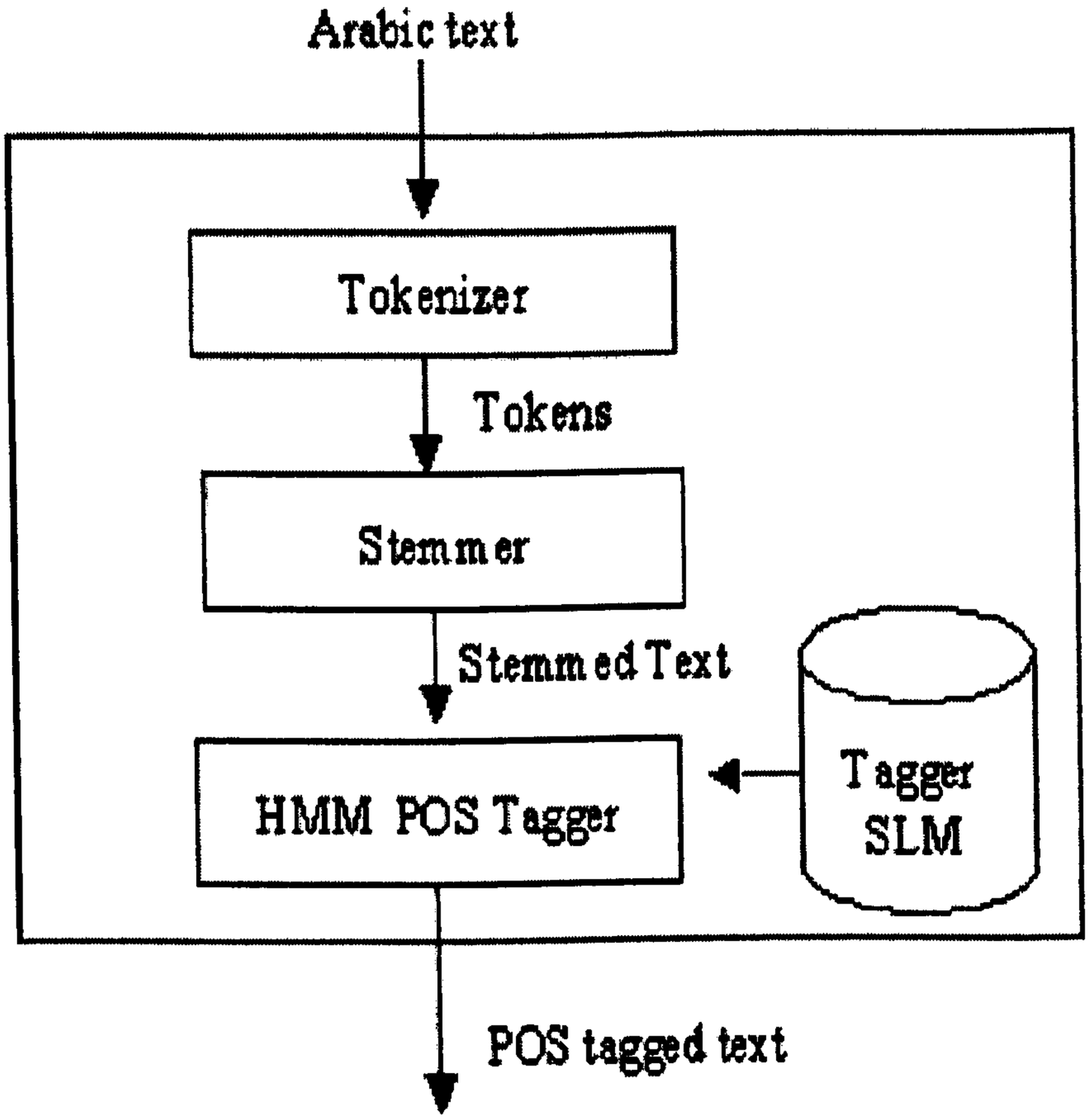

Figure 02.14: The architecture of Shamsi and Guessoum's system (Shamsi and Guessoum, 2006)

Generally, this is not accurate and this lead to incorrect results as illustrated in Table 2.8.

Table 02.8: Incorrect tagging of the word Salem

The Arabic phrase	***The English translation***
ذهب سالم إلى المكتبة	Salem went to the library

No doubt that the correct tag of the word (Salem, سالم) in this example is a proper noun According to this incorrect tag all the result of Shamsi and Guessoum's system which is relying on this tag will be inaccurate.

```
Initializing in-memory dictionary handler.
Loading dictionary : dictPrefixes .
78 entries totalizing 299 forms
Loading dictionary : dictStems .....
38600 lemmas and 47261 entries totalizing
82158 forms
Loading dictionary : dictSuffixes ..
206 entries totalizing 618 forms
Loading compatibility table : tableAB ...
1648 entries
Loading compatibility table : tableAC .
598 entries
Loading compatibility table : tableBC ..
1285 entries
... done.
Initializing in-memory solutions handler.
... done.
possible analysis
سالم of the input word =VERB_IMPERFECT
possible analysis
of the input word سالم =NOUN_PROP
```

Figure 02.15: The output of Buckwalter for the word (Salem, سالم).

Consider the example shown in Table 2.9 the correct tag of the word (Dubai, دبي) is proper noun (city name).

Table 02.9: An example showing an incorrect tag for the word Dubai

The Arabic phrase	***The English Translation***
غادر من دبي	Departure from Dubai

However the first output of the Buckwalter for the word (Dubai, دبي) as shown in Figure 2.16 is noun. Consequently, the results which they obtained will be also incorrect.

However we have examined numerous words with Buckwalter morphological analyzer, and we have found the correct tag of the majority of them are not the first output, therefore this approach is not applicable.

```
Initializing in-memory dictionary handler.
Loading dictionary : dictPrefixes .
78 entries totalizing 299 forms
Loading dictionary : dictStems .....
38600 lemmas and 47261 entries totalizing
82158 forms
Loading dictionary : dictSuffixes ..
206 entries totalizing 618 forms
Loading compatibility table : tableAB .
1648 entries
Loading compatibility table : tableAC .
598 entries
Loading compatibility table : tableBC .
1285 entries
... done.
Initializing in-memory solutions handler.
... done.
possible analysis
of the input word دبي =NOUN
possible analysis
of the input word دبي =ADJ
possible analysis
of the input word دبي =NOUN
possible analysis
of the input word دبي =NOUN_PROP
```

Figure 02.16: The output of Buckwalter for the word (Dubai, دبي).

Furthermore, the systems adopted statistical approach required a large amount of training data to improve the system performance. However it is essential to have someone to make ready the training data (corpus), a corpus of texts needs to be collected and further processed properly for the information to be extracted. Once a suitable training corpus is available, a training algorithm is needed to run over the corpus to derive statistics that are used to build up a trained model that the system can employ in analyzing and processing new data (Eikvil, 1999). In addition any word in the corpus given an incorrect tag will

certainly have a massive impact on the overall precision of the system. Benajiba, Rosso, & Benedi, (2007) presents a NER system adjusted to the Arabic language, they adopted statistical approach namely maximum entropy. However their experimental lack of very crucial tool i.e. part of speech tagger. Moreover they used a limited number of articles as a training corpus.

2.4.3 Classification of Arabic Named Entity Systems

As we have seen in the previous sections the systems used in the Arabic field, either adopted a rule based or statistical approach, applied into two types of text, transliterated text and raw text. We have summarized these systems as shown in table 2.10 depending on the approaches which have been adopted and the type of the text has been applied.

Table 02.10: Classification of some systems depending on the approach and the text applied

	Transliteration Arabic	***Raw Arabic***
Rule Based	▪ Samy et al. (2005)	▪ Saleem (2004) ▪ Shaalan and Raza (2007)
Statistical	▪ Larkey *et al.* (2003) ▪ Zitouni *et al*, (2005) ▪ Benajiba and Rosso (2008)	▪ Shamsi and Guessoum (2006) ▪ Benajiba et al (2007)
other	▪ Pouliquen *et al.* (2005) ▪ Kashani et al (2006)	

Using transliterated Arabic text into the NER systems presents some challenges, these challenges can be summarized as a follows:

- Arabic lack some sounds and letters from the other language. For instance, there is no perfect match for "ع" in English this leads to ambiguities in the process of transliteration.
- The omission of diacritics and vowels in almost all the Arabic writings. Diacritics are considered to be one of the main causes of ambiguity when dealing with Arabic proper nouns.
- The existence of common Arabic mistakes in which different characters are used interchangeably; like Hamza errors (ا ، إ ، أ ، آ) Yaa errors (ى ، ي), and Taa Marbuta errors (ة ، ه).
- Most of the systems uses the transliteration text recognises only the proper noun and excludes the rest of the named entity.
- Generally the systems rely on the transliteration text, for instance (Samy et al, 2005) require a corpus in the target language containing the correct transliteration.

2.4 Summary

In this chapter we provided an overview of the field of Information Retrieval in general and Information Extraction in particular and we have summarised the development in the field. We found that IE has four main tasks and one of the most important of these tasks is Named Entity recognition which is the topic of this thesis. There are two basic approaches to IE system development: the rule based Approach and the automatic training approach. This was followed by a short review of some application domains of IE for the English

language. The chapter also reviewed the development of the IE field for the Arabic language. We first started by summarizing the challenges faced by IE systems for the Arabic language and the limitations Arabic NLP resources. This was followed by a thorough review of NER systems in Arabic. The chapter concluded by summarizing and classifying the research efforts in the NER field in Arabic. The next chapter describes in details the characteristics of the Arabic language.

Chapter 3

Introduction to the Arabic Language

3.1 Introduction

The Arabic language is the sixth most widely spoken language in the world. More than 300 millions of people over the world are using the Arabic language as their first or second language. It is the official language in 22 countries. Unlike the Indo-English languages, Arabic is written from right to left and most of the letters in an Arabic word are connected with each other except very few letters which can stand alone. Additionally, there is no special feature in Arabic language such as a capital letter in the beginning of a word to indicate for instance proper names such as the names of people, and countries as is the case in languages such as English and French. This fact increases the ambiguity in the text and as a result, tasks such as Information Extraction (IE) generally and Named Entity Recognition (NER) in particular are more complicated in Arabic.

The Arabic Language can be classified into three types: Classical Arabic (العربية الفصحى), Modern Standard Arabic (العربية الحديثة), and Colloquial Arabic dialects (العربية العامية). Classical Arabic is the language of the holy Quran. It could be also viewed as the language of the pre-Islamic poets. This language is fully vowelized and is rarely used in today's everyday writing. Modern Standard Arabic (MSA) is the language of today's Arabic newspapers, magazines, periodicals, letters and modern writers. It is also used as the medium of oral communication in formal speeches and in television and radio broadcasts. MSA could be viewed as classical since there have been no major changes modifying the structure of the classical language. MSA, however, differs from Classical Arabic in two aspects: adopting minor stylistic changes and expanding the lexicon to include new

technical terms (Aljohar, 1999). Colloquial Arabic dialects, on the other hand, consist of the languages of the different Arab countries. There are two main classes of Arabic dialects: the Eastern dialects of Egypt, Sudan, and the Middle East, and the Western dialects of North Africa (Tomokiyo, 2003). These dialect classes are distinguished by the reduction of the vowel system in the Eastern dialects and a contrast in the stress system (Janet, 2002). Dialects are used for every-day oral communications by the people of the Arab dialect areas. There are no written transcripts for such dialects (Harmain and Aljohar, 2001).

The Arabic language belongs to the Semitic languages group that includes the Hebrew language. Arabic differs from Indo-European languages syntactically, morphologically and semantically (Sabri et al, 2006). The Arabic alphabet contains 28 letters, 25 are consonant, and the remaining 3 are used as long vowels. These are the same as the "ea" in beak, the "oo" in food, and the "a" in dad. The Arabic alphabet has a special feature that is the huge number of diacriticals points and slashes above and under the letters. Moreover, these diacriticals are usually used to distinguish between consonants. In addition Arabic script is complicated due to the context sensitivity of its characters' writing. The majority of the letters have four appearances depending on their position in the word (beginning, middle, end and separate).

Table 3.1 gives an example of the different forms of the letter "gh" in different positions.

Some letters of the Arabic alphabet look similar in shape but are distinguished from one another by the position of small dots called "diacritical points" (نقطة) "Nuqta". For example ب ت ث have the same shape, but ب has one dot below, ت and ث have two and three dots above respectively. ج, ح and خ are differentiated from one another by the position of the dot or the absence of it. The same is true for د and ذ, ر and ز, س and ش, ص and ض, ط and ظ

and ع and غ. letter ى y is written with two dots below or without dots. Moreover there are some Arabic consonants which have no equivalent in the English alphabet.

Table 3.1: The variation of the letter (gh)[1].

Beginning	***Middle***	***End***	***Separate***
غـ	ـغـ	ـغ	غ

On the other hand the Arabic language had originally no signs for the short vowels; however vowels are usually not indicated in Arabic books, journals or in any written matter, unless when the correct pronunciation of a word is to be made sure. A person who knows Arabic grammar can read correctly any Arabic text without the help of vowel sign forming part of the text, but non Arabic or beginner readers very much needs them. Every consonant in a vocalized Arabic text is provided with a vowel sign (sometimes indicated and often omitted in writing). There are three short vowels in the Arabic language; (اَلْفَتْحَةُ, fathah) is a small diagonal stroke above a consonant, it is indicated with vowel "a" in English word "man", (اَلضَّمَّةُ, dammah) which is a small sign above a consonant It is identical with vowel "u" in the English word "bull", and (اَلْكَسْرَةُ, kasrah) a small diagonal stroke under a consonant, it is identical with "i" in the English word "finish".

3.2 The Morphology of the Arabic Language

The term *morphology* comes from the antique Greek (*morphe*) and means *shape* or *form*. The general definition of morphology is "the study of form or pattern", i.e. the shape and arrangement of parts of an object, and how these "conform" to create a *whole* or Gestalt (Ritchey, 1998). The morphological analyzer retrieves for a given word all its possible readings as shown in Table 3.2.

---

(1) We used the Qalam transliteration scheme to present any Arabic word or sentences throughout our thesis. The transliteration chart is found in Appendix A.

Table 03.2: Morphological analysis of the word قبل

Word	***Transliteration***	***POS Tag***	***Meaning***
قبل	qabla	Preposition	Before
	qabila	Perfect Verb	(he) accepted
	qubila	Passive form, past tense	(it-masculine) was accepted
	qabbala	Perfect Verb	(he) kissed

The Arabic language has a rich vocabulary and a complex morphology. The morphological representation of Arabic is rather complex because of the morphological variation and the agglutination phenomenon (Kadri and Benyamina, 1992). Arabic is a highly inflectional language with 85% of words derived from trilateral roots. Moreover, there are around 10.000 independent roots (Al- Fedaghi and Al-Anzi, 1989). Arabic roots are surrounded by a huge number of prefixes, suffixes, or both. Furthermore the prefix and suffix could be associated to any type of Arabic word such as noun, verb, adjective etc. The prefix and suffix are considered as an indicator of the grammatical category of the word such as person, numbers and gender as illustrated in Table 3.3

Table 03.3: A sample of Arabic affixes

Antefixes	***prefixes***	***suffixes***	***postfixes***
وبال,وال,بال,فال,كال,ولل, ال,وب,ول,لل,فس,فب,فل, وس,ك,ف,و,ب,ل	ا,ن,ي,ت	تما,يون,تين,ات,تان,ان ,ون,ين,وا,تا,تم,تن,نا,ت,ن,ا ,ي,و	كما,هما,كن,هن,تي, ها,نا,هم,كم, ك,ه,ي
Prepositions meaning respectively: and with the, and the, with the, then the, as the, and to (for) the, the, the, and with, and to (for), then will then with, then to (for), and will, as, then, and, with, to (for)	Letters meaning the conjugation person of verbs in the present tense	Terminations of conjugation for verbs and dual/plural/female marks for nouns	Pronouns meaning respectively: your, their, your, their, my, her, our, their, your, your, his, my

Moreover, prefixes can be combined with each other; accordingly the result will be compound prefixes, and the same can be occurring to the suffixes, Table 3.4 shows an example for the word (negotiate, فاوض) combined with all types of affixes.

Table 3.4: An agglutinated form of the Arabic word "لیفاوضونهم, to negotiate with them"

antefix	*prefix*	*core*	*suffix*	*postfix*
ل	يـ	فاوض	ون	هم
Preposition meaning "to"	A letter meaning the tense and the person of conjugation	negotiate	Termination of conjugation	A person meaning "them"

In addition the root itself can be extracted from the stem, as shown in Table 3.5.

Table 03.5: Root extraction from the stem "enter, أدخل".

	The stem	***The root***
Arabic meaning	أدخل	دخل
Translation	Enter	Entered

In Semitic languages like Arabic the majority of nouns, verbs and adjectives are derived from a known set of roots to which infixes are added as shown in Table 3.6.

Table 03.6: A sample of words extracted from the root (write/كتب).

The word	***prefix***	***suffix***	***Translation***
أكتب	أ	-	Write
يكتب	ي	-	He write
تكتب	ت	-	She write
نكتب	ن	-	We write
أكتبوا	أ	وا	Write (for masculine and feminine)
يكتبان	ي	ان	They write (for dual masculine)
تكتبان	ت	ان	They write (for dual feminine)
يكتبون	ي	ون	They write (for masculine)
يكتبن	ي	ن	They write (for feminine)

Text in Arabic can be written with or without diacritic indicators. A word like (درس) (without diacritic indicators), will have different meanings such as (study/درس) and (lesson/ درس) etc. Certainly, the diacritic text is more adjusted for the human reader as it is

easy to read and pronounce. Unfortunately, the majority of Arabic written texts in recent years such as magazines, newspapers and electronic texts do not include diacritics. Diacritic indicators are only used in some specialized books such as children, religious and poetry books. However, a morphology system is the backbone of NLP. No application in this field can survive without a high-quality morphology system to assist it.

The Arabic language has its own characteristics that are not found in other languages, i.e. the extremely complex morphology. This has made it a very challenging research area. Hegazi and ElSharkawi (1986) produced a system to identify the root of any Arabic word along with morphological patterns and word categories. The system has also been used for recognition and adjustment mistakes in spelling and vowelization. Al-Fedaghi and Al-Anzi (1989) developed an algorithm to generate the root and the pattern of a given Arabic word, the core concept in the algorithm is to locate the position of the three letters of a possible triliteral root in the pattern and check to see whether the candidate trigram appears in a list of known roots. El-Sadany and Hashish (1989) developed an algorithm designed to perform both analysis and generation and was able to deal with vowelized, semi-vowelized, and non-vowelized Arabic words.

Al-Shalabi and Evens (1998) developed a system that removes the longest possible prefix from the word where the three letters of the root must lie somewhere in the first four or five characters of the remainder, then he generates some combinations and checks each one of them with all the roots in the file. Roeck and Waleed Al-Fares (2000) developed a clustering algorithm for Arabic words sharing the same verbal root; they used root-based clusters to substitute for dictionaries in indexing for information retrieval. Beesley and Karttunen (2000) developed a new method for constructing finite-state transducers that

involves reapplying a regular-expression compiler to its own output. They implemented the system using an algorithm called compilereplace.

Berri (2001) created a system consisting of three main modules: a rule knowledge base which has the regular and irregular morphological rules of the Arabic grammar, a set of word lists containing the exceptions handled by the irregular rules, and a matching algorithm that matches the words to the rules. Darwish (2002) presented a morphological analyzer called Sebawai, this analyzer identifies the prefix, stem and suffix parts for a given word and uses morphological rules for deriving roots from stems. Buckwalter (2004) developed a morphological analyzer and this will be described in details in Chapter 4 since we used this analyzer in our system.

3.3 Arabic Stemming

A stemming algorithm is a computational procedure which reduces all words with the same root (or, if prefixes are left untouched, the same *stem)* to a common form (Lovins, 1968). One of the simplest stemming techniques used in English and many other western European languages is by stripping each word of its derivational and inflectional suffixes. In this approach lists of suffixes are used to reduce words to their bare form. The most common stemming algorithms for English are Porter (Porter, 1980) and Lovins (Lovins, 1968). Kraaij (1996) concluded that stemming improves recall. Despite the fact that the Arabic language is very difficult to stem, stemming appears to have more positive effect when the language is highly inflected (Popovic and Willett, 1992). However, there are significant works done on stemming Arabic using various approaches. Early work on Arabic stemming used manually constructed dictionaries. Al-Kharashi and Evens (1994) and Abu-Salem (1999) worked with small text collections, for which they manually built

dictionaries of roots and stems for each word to be indexed. The affix removal approach is generally called light stemming when applied to Arabic, referring to a process of stripping off a small set of prefixes and/or suffixes, without trying to deal with infixes, or recognize patterns and find roots, such as (Aljlayl, 2001) and (De Roeck, 2000). Chen and Gey (2002) used a parallel English-Arabic corpus and an English stemmer to cluster Arabic words into stem classes based on their mappings to English stem classes. Rogati (2003) uses a statistical machine translation approach that learns to split words into prefix, stem, and suffix by training on a small hand annotated training set and using a parallel corpus. Rule-based stemming for Arabic is a problem studied by many researchers; an excellent overview is provided by (Larkey, 2002). In our application, we adopted the Buckwalter's stemmer (Buckwalter, 2004) which returns all valid segmentations based on the fact that an Arabic prefix length can go from zero to four characters, the stem can consist of one or more characters, and the suffix can consist of zero to six characters. All the valid segmentation solutions are then passed through to the POS tagger.

3.4 Arabic Parts of Speech

POS tagging is the process of assigning a part-of-speech tag such as noun, verb, pronoun, preposition, adverb, adjective or other tags to each word in a sentence. It reflects the word syntactic category based on its context for the purposes of resolving lexical ambiguity (Jurafsky, 2000). There are three main approaches for the parts of speech tagging that can be classified as:

- Linguistic approach: consists of coding the essential knowledge in a set of rules written by linguists.
- Statistical approach: needs a few amount of human effort; however in the recent year this approach has been used successfully particularly hidden Markov Models.

- Learning algorithms: obtains a language model from a training corpus and uses an example-based learning method and a distance measure to make a decision on which of the previously learned examples is more similar to the word to be tagged.

In the last decade, tagging has been one of the most interesting problems in the natural language learning community (Andrew, 2003). It is the essential basic tools required in speech recognition, parsing, information retrieval, information extraction, and developing language corpus. The majority of the words in the text have more than one morphological analysis. The responsibility of POS tagger is to assigning each word with the most suitable morphological tag. The word marked with a bold font in Table 3.7 represents the correct POS tag for each word in the sentence.

Table 03.7: Part-of-Speech tagging for the sentence كتب فريد قبل وسيم

Word	*Transliteration*	*POS Tag*	*Meaning*
كتب	kataba	**Verb**	**He wrote**
	kutub	Noun	Books
فريد	Farid	**Proper Noun**	**Farid**
	Farid	Adjective	Unique
قبل	qabla	**Preposition**	**Before**
	qabila	Perfect Verb	(he) accepted
	qabbala	Perfect Verb	(He) kissed
	qubila	Passive form, past tense	(It-masculine) was
وسيم	Wasym	**Proper Noun**	**Wasym**
	Wasym	Adjective	beautiful

Arab grammarians have divided the parts of speech into three categories, (noun, أسم) (verb, فعل) and (particle,حرف). In the Arabic language there are a number of signs that point out whether a word is a noun or a verb. One of them is the affix of the word: some of the affixes are used with verbs; some of them are used with nouns; and some of them are used with verbs and nouns, for instance verbs can be attached to a prefix indicating the future (س, s / will) or preceded by a word indicating the future (سوف, swfa / will) and word preceded by a definite article (ال, the) must be a noun. The patterns are another sign to

distinguish the type of the word; some of these patterns are used just for nouns; some of them are used just for verbs; and others are used for both. One more sign comes from grammatical rules; several grammatical rules can be used to distinguish between nouns and verbs, some letters in the Arabic language mark the nouns; others mark the verbs.

3.4.1 Singular, Dual and Plural in Arabic Language

In Arabic language there are three numbers, namely: (اَلمُفْرَد, the singular, اَلمُثَنَّى, the dual and the plural, الجَمع).

- (اَلمُفْرَد ,The singular): a word indicating one thing only e.g. (بَابٌ ,door)
- (اَلمُثَنَّى, the dual): a word indicating two things. In order to formulate the dual, (آن, an) is suffixed to the singular word in the nominative case and by (ين, yn) in the accusative and genitive cases. This variety of the dual depends on its position in the statement. In other languages like English the dual take one situation as shows in the example given in Table 3.8.

Table 03.8: The position of the dual in Arabic

The word	***The position***	***The translation***
الولدان	nominative	Two boys
الولدين	Accusative and genitive	Two boys

- (الجمع, the plural): there are two kinds of plurals in Arabic; the first is (الجمع السالم the sound plural): When a plural retains all the vowels and consonants of the singular and is based on its pattern. This plural consists of two types, (جمع المذكر السالم, Masculine sound plural): it is suffixed by (ون, wn) if it is Nominative and it is suffixed (ين, yn) if it genitive and (جمع المؤنث السالم, Feminine sound plural): it is suffixed by (ات, at) at the end of a singular feminine noun, either it is definite or indefinite. All these kind of plurals are shown in Table 3.9.

Table 3.9: The position of the plural in Arabic

The word	***The plural***	***The position***	***The translation***
معلم	معلمون	Masculine sound plural / nominative	The teachers
معلم	معلمين	Masculine sound plural / accusative and genitive	The teachers
معلم	معلمات	Feminine sound plural/ in all cases	The teachers

The second type of the plural is (جمع التكسير, the Irregular plural): it varies very much from its singular as the example given in table 3.10.

Table 3.10: An example of an irregular plural

The singular	***The Irregular plural***
(إمرأة, woman)	(نساء, women)

3.4.2 Nouns and Noun phrases

Arabic nouns can be subcategorized into adjectives, proper nouns and pronouns, and the nouns are divided into two categories, (مذكر, masculine) and (مؤنث, feminine). The majority of the nouns ending with the letter (ة, t) are feminine as shown in Table 3.11. The nouns also have three persons, one to depict the speaker (first person), one to depict the person being addressed (second person) and one to describe the person that is not presentd person).

Table 3.11: Example of the differentiation between the masculine and feminine.

The word	***Masculine***	***Feminine***
الطفل The child	طفل child	طفلة child
الطالب The student	طالب Student	طالبة Student
الطبيب The doctor	طبيب doctor	طبيبة doctor

There are two kinds of phrases in the Arabic language, noun phrases (NP) and verb phrases (VP). The general form of a noun phrase is: NP →Subject, Predicate, as shown in the example given in Table 3.12.

Table 03.12: An example of a noun phrase.

The phrase	***The translation***
الباب مفتوح	The door is open.

The noun phrase usually begins with a pronoun or a noun. However, this name represents a specific entity such as person, animal, sun, city etc. This is the first part of the noun phrase; the second part represents specific information about the subject called Predicate. A predicate might be a noun, an adjective, or a preposition. Noun phrases sometimes start with a pronoun followed by a name. Hence these pronouns can be used as a clue to guide us to identify named entities (person's name). Examples are shown in Table 3.13.

Table 03.13: Examples of noun phrases ending with a noun.

The phrase	***The translation***
أنا علي	I am Ali
هي سالمة	She is Salema
هو سالم	He is Salem

The noun phrase can be definite or indefinite. The definite state is marked by the article (ال, al / the) as in (الطبيب/ the doctor). Otherwise, the noun phrase is an indefinite.

3.4.3 The Annexation Noun

The annexation considered as a nominal expression consists of two nouns combined together to give meaning to one phrase or a complex phrase. The first part of this expression is usually isolated from a definite article. The second part, which follows the first word of this expression, will probably begin with a definite article. In addition the case of the declension of the second word is always genitive. Accordingly, some rules have

been generated and used in our system to obtain a named entity from the annexation phrase, we considered the first word as a clue, and then probably the second word is a named entity as shown in Table 3.14.

Table 3.14: Examples of named entities obtained by annexation.

The phrase	*The translation*
نهر النيل	The Nile river
كتاب محمد	Mohammed's book
سوق طرابلس	Tripoli market
مطار هيثرو	Heathrow airport
جيش المهدي	Al Mahdi army
إدارة بوش	Bush administration
سفير ليبيا	Libya ambassador

3.4.4 Verb Phrases

The Verb is any word referring to an action in a certain time; it can be perfect, imperfect or imperative. In general Verbs are almost the same as in English. Perfect verbs are used to describe completed actions; imperfect verbs indicate uncompleted actions; while imperative verbs express an order. The verb phrase must begin with a verb, and the general form of the verb phrase is: (Verb) (Optional subject) (Optional object), as shown in the example given in Table 3.15. In this example all the elements of the verb phrase exist. Moreover, the statement is arranged in the normal form, verb, subject, and object respectively.

Table 3.15: Example of a verb phrase where the verb is in the normal position.

The phrase	*The translation*
كتب محمد الدرس	Mohammed wrote the lesson.

The arrangement of the subject and the object does not always follow this form. Sometimes the subject in the phrase comes before the object. Moreover, the subject or the object might be omitted from the verb phrase, since they are not required in the phrase.

However, it depends on the general logic of the phrase. In the example given in Table 3.16, the subject is omitted, because from the context we know that there is a subject in the statement despite it is not visible.

Table 3.16: An example where the subject is omitted.

The phrase	***The translation***
أفهم الإنجليزية	I understand English

On the contrary of the normal position, the example given in Table 3.17 the object (the money, المال) established in the phrase before the subject (Salem, سالم).

Table 3.17: An example where the object comes before the subject

The phrase	***The translation***
أخذ المال سالم	Salem took the money

However, many techniques have been used to tag English and other European languages corpora. The first technique developed was the rule-based technique used by Greene and Rubin (1971). They used context-frame rules to select the appropriate tag for each word; their system achieved an accuracy of 77%. A rule-based tagger has been developed by Brill (1994) and achieved an accuracy of 97.5%. Statistical and hybrid part-of-speech taggers have been used very successfully for English; hidden Markov models were used to develop Part-of-Speech taggers, which achieved an accuracy of 97% (Garside, 1987). Church's PARTS tagger (Church, 1988) and the Xerox tagger, which was developed by Doug Cutting, achieved an accuracy of 96% (Cutting, 1992). Neural networks have been used in developing Part-of-Speech taggers, Marques (1996) developed a tagger for Portuguese that achieved an accuracy of 96%. However, there are insignificant works done in POS tagging for Arabic such that developed by Freeman (2010). A framework of a

hybrid Arabic POS tagger has been introduced in (Khoja, 2001) without specifying a particular statistical method. A Support Vector Machine (SVM) based POS tagger was described in (Diab et al., 2004). A Hybrid method has been developed by Yamina (2006) which learns tagging Arabic by a combination of based-rules and a memory-based learning.

3.5 Summary

In this chapter we provide an overview of the characteristic of the Arabic language, and we concluded that the Arabic language has its own features and characteristics particularly with regards to morphology. The Arabic language has a very complex morphology because of the derivational and inflectional nature of the language. We gave a brief review of some of the most popular morphological analysis systems used in the field. As a result of Arabic being a highly inflected language, stemming appears to have more positive effect in this case and we reviewed several stemmers used for the Arabic language. Finally, we reviewed some part of speech taggers used to tag the Arabic language. These taggers adopted various approaches i.e. linguistic approach, statistical approach and learning algorithms.

Chapter 4

The Architecture of the Arabic Information Extraction System

4.1 Introduction

The General Architecture of Text Engineering (GATE) is one of the most popular software dealing with NLP. GATE is an infrastructure for developing and deploying software components that process human language (Cunningham et al, 2002). The motivating factors behind the development of GATE included the facilitation of reuse of components, comparative and task-based evaluation, collaborative research, software-level robustness, efficiency and portability (Cunningham et al, 1995). Moreover, GATE provides a set of NLP tools including tokeniser, gazetteer, POS tagger, chunker and parsers which are essential tools for any development of natural language systems. GATE is heavily used in research and teaching of NLP and is freely available (Cunningham, 2000). Hence, to exploit GATE features to achieve our research objectives, we included the GATE in our system. In this chapter, we will discuss in details the GATE system and the Buckwalter Arabic Morphological Analyzer (BAMA), as both of these systems are used in our application.

4.2 The General Architecture for Text Engineering

GATE was developed at the University of Sheffield in 1996 and is well-known in the NLP field and is used in various NLP applications such as information extraction (IE). GATE has been used for many IE projects in many languages and problem domains, such as the Large Scale Information Extraction (LaSIE) project. LaSIE performs named entity recognition, coreference resolution, template element filling and scenario template filling

(Gaizauskas, 1998). Advanced Information System for Multinational Drug Enforcement (AVENTINUS) is an EU funded research and development programme set up to build an information system for multinational drug enforcement (Cunningham et al, 2000).

4.2.1 GATE Architecture

GATE is composed of three significant subsystems as shown in Figure 4.1:

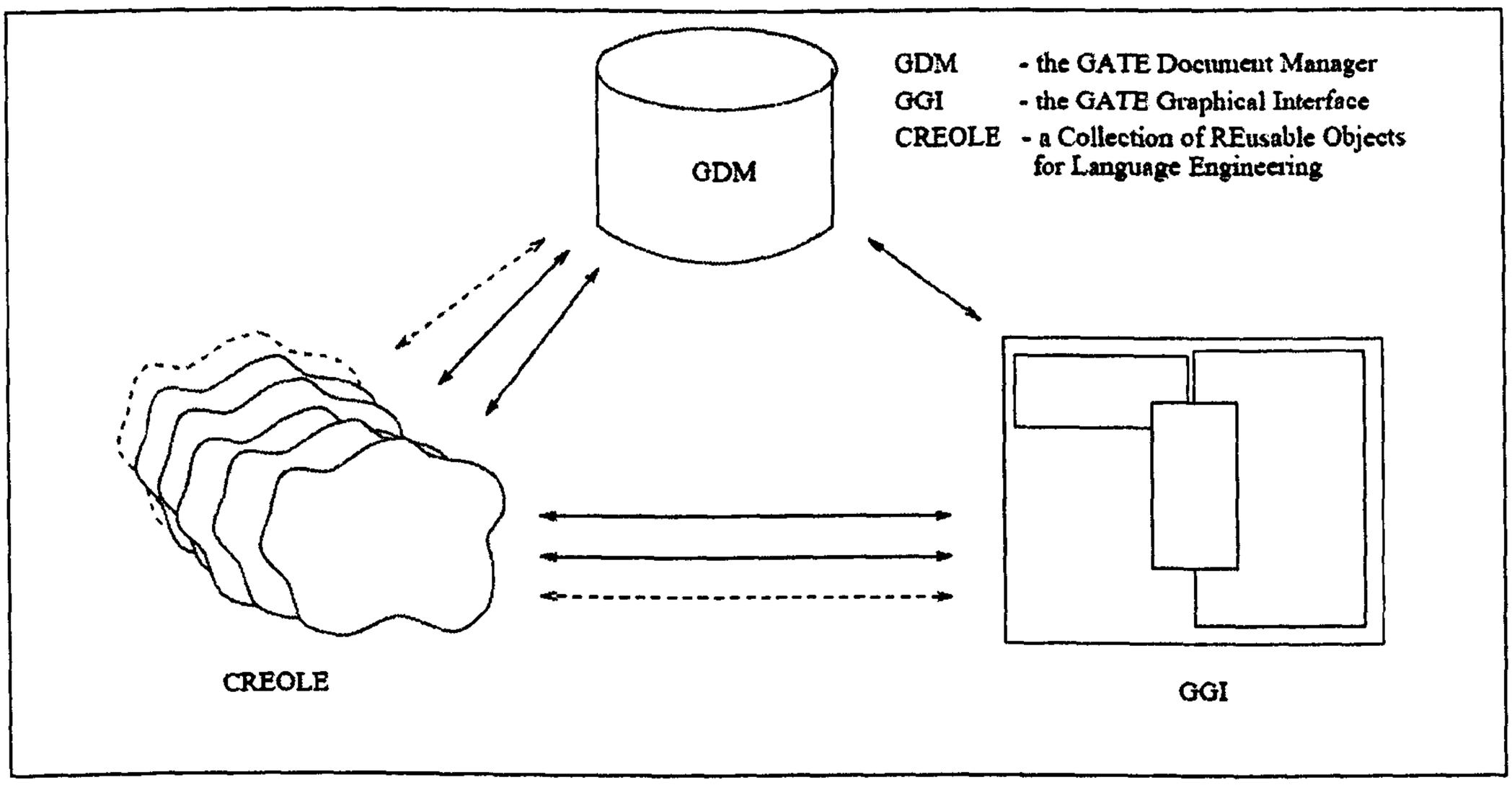

Figure 4.1: The GATE Architecture (Cunningham, 2000)

- GATE Document Manager (GDM) is a database for storing information about texts such as lexicons, corpora, documents, etc. All communication between the components of a Language Engineering (LE) system goes through GDM. One of the key benefits of adopting an explicit architecture for data management is that it becomes possible to easily add a layer graphical interface access to architectural services and data visualisation tools.

- GATE Graphical Interface (GGI) is a development tool providing integrated access to the services of the other components and adding visualisation and debugging tools. GGI as shown in Figure 4.2 has functions for creating, viewing and editing collections of documents (corpora) which are managed by the GDM. GGI also has facilities to display the results of module or system execution.

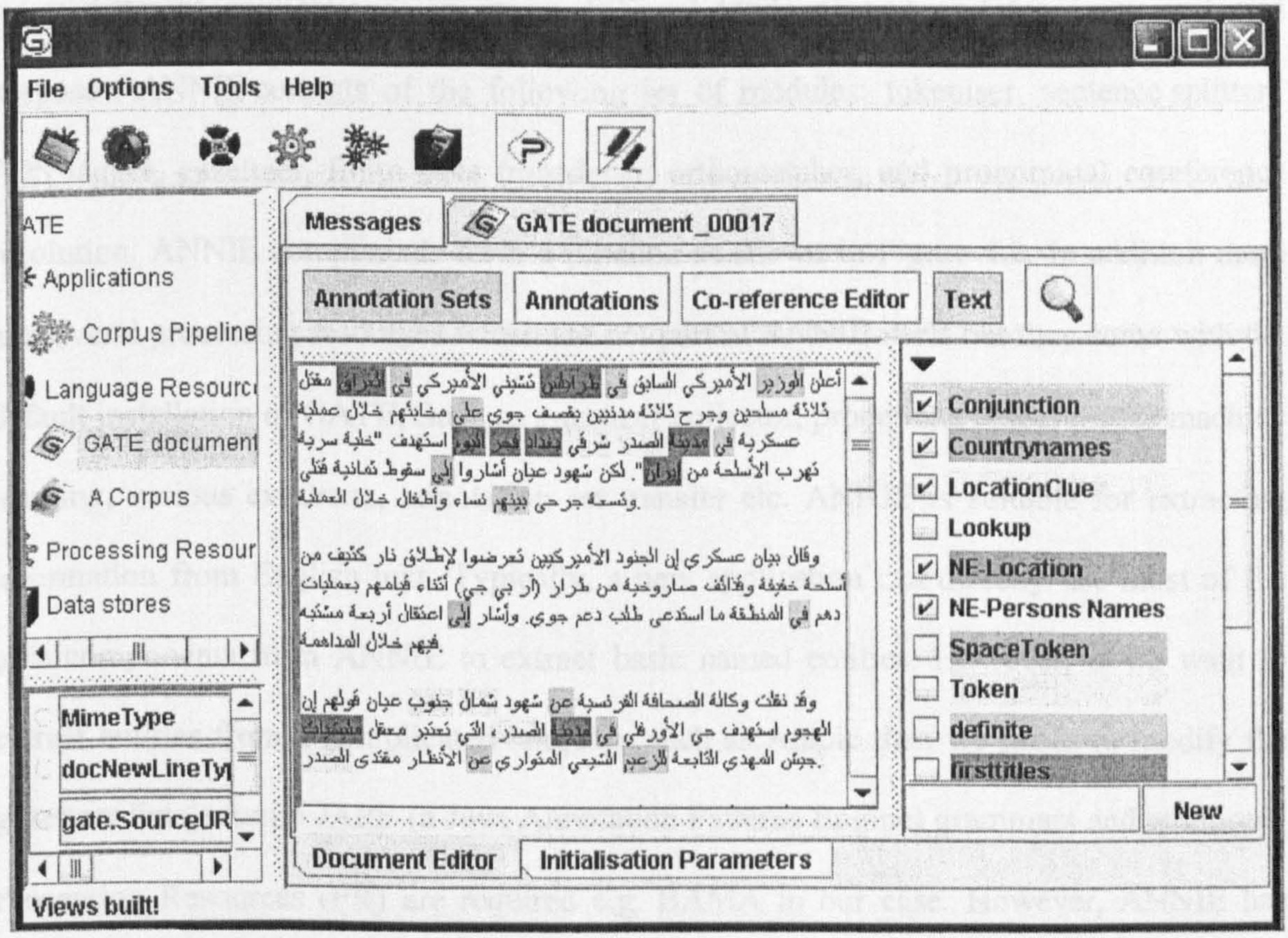

Figure 4.2: The GATE graphical interface

- CREOLE (a Collection of REusable Objects for Language Engineering) a set of Language Engineering (LE) components integrated with the system. A CREOLE module does all the real work of processing texts and discovering information about their content. Usually, a CREOLE object will be wrappers around a pre-existing LE database, for instance a tagger, parser, sentence splitter, etc.

4.2.2 Information Extraction System within GATE

GATE is distributed with an Information Extraction component set called ANNIE (A Nearly-New IE) system. ANNIE is provided as part of GATE (Cunningham et al, 2002), which is an architecture, framework and development environment for language processing research and development. In other words ANNIE is intended to be useable in many different applications, on many different kinds of text and for many different purposes. ANNIE consists of the following set of modules: tokeniser, sentence splitter, POS tagger, gazetteer, finite state transducer, orthomatcher, and pronominal coreference resolution. ANNIE components form a pipeline as shown in Figure 4.3. In addition there are several processing resources which are not part of ANNIE itself but they come with the default installation of GATE, Such as gazetteer collector, processing resources for machine learning, various exporters, annotation set transfer etc. ANNIE is suitable for extracting information from English text. Typically, a new application can directly use most of the core components from ANNIE to extract basic named entities. However, if we want to extract entities from a complicated language such as Arabic then we needs to modify the gazetteer lists, rewrite JAPE (a Java Annotation Patterns Engine) grammars and additional Processing Resources (PR) are required e.g. BAMA in our case. However, ANNIE has very good performance (F measure 92.9%) for traditional information extraction on general news texts (Maynard et al, 2007) and therefore it is a good base to build on.

4.2.3 Processing Resource

As we mentioned above, one of the most important resources in GATE is the Processing Resource (PR). In the NLP field, the researchers required the Processing Resource to examine and evaluate their assumptions; consequently the availability and accessibility of these resources are essential.

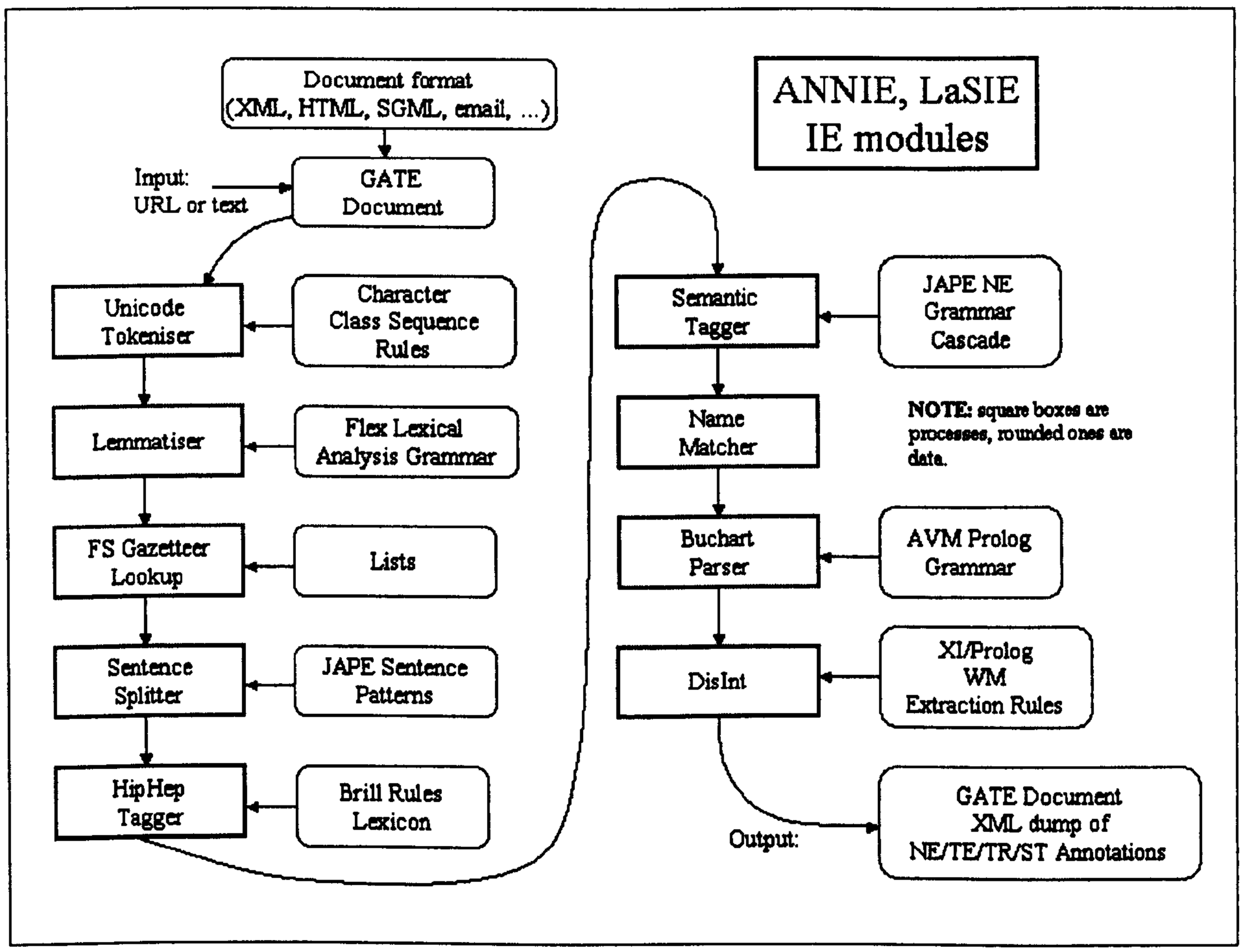

Figure 4.3: The workflow of ANNIE (Cunningham et al, 2010)

However, the developer (language engineer) is able to use either the Processing Resource which is already embedded with GATE or create his own Processing Resources and integrate them into GATE through the plug-and-play method, and GATE contains a number of these resources.

4.2.3.1 The Tokeniser

The tokenization is the process that analyses and split the input text into a number of tokens such as, word, number, symbol, space, etc. The system will take the raw text from the collection of documents (corpora), and tokenize each document. A word is considered as a sequence of connected letters either upper or lower case, a number as sequence of

digits, a symbol represented as @, #, etc. The rest of the tokens are considered as a gap between words and are represented as a white spaces as shown in Figure 4.4.

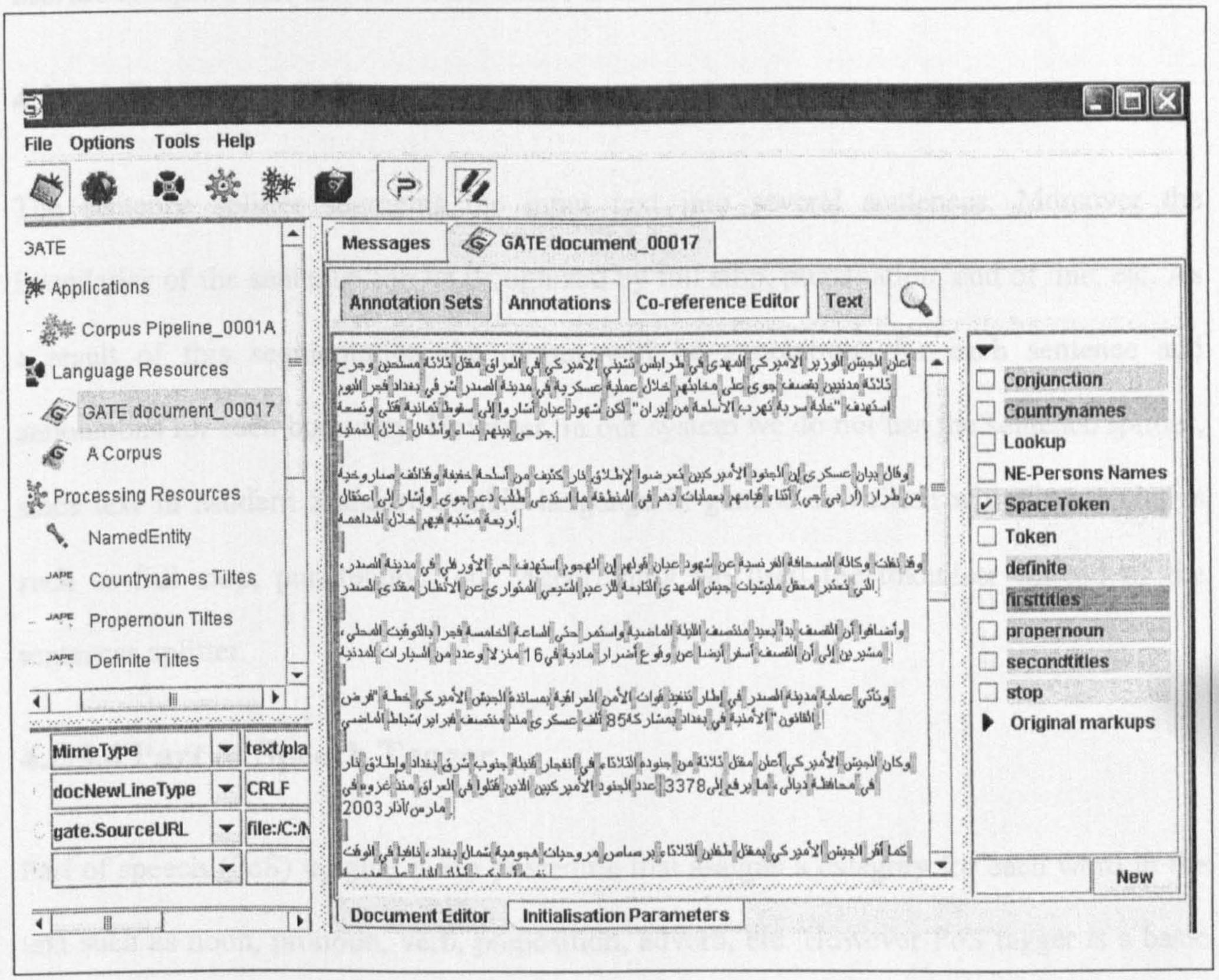

Figure 04.4: An example of a text after tokenization.

4.2.3.2 Gazetteer

The gazetteer consists of lists storing specific information such as people's names, organizations names, locations names, days of the week, etc. These are usually used when the number of instances of a particular class of named entities is finite and could be stored in a database. For example it is easy to identify the days of the week in text by referring to

an existing list rather than writing complex rules to identify these entities. Gazetteers can also store lists of keywords that can help identify some entities within documents. GATE offers a list of gazetteers and allows the creation of user defined gazetteers. The gazetteer lists are compiled into finite state machines to be able to match the text tokens.

4.2.3.3 Sentences Splitter

The sentence splitter segments the input text into several sentences. Moreover the boundaries of the sentence can be recognized by full stop, punctuation, end of line, etc. As a result of this segmentation the output will be annotations for each sentence and annotations for each boundary. However, in our system we do not use the sentence splitter, since text in Modern Standard Arabic language in general is written without boundaries such as full stop, punctuation, etc. Accordingly we used the tokeniser instead of the sentences splitter.

4.2.3.4 Part of Speech Tagger

Part of speech (PoS) tagging is the procedure that assigns a category for each word in the text such as noun, pronoun, verb, preposition, adverb, etc. However PoS tagger is a basic tool for various applications in NLP field such as IR, IE, etc. Moreover, POS tagger is necessary as a tool to build up any language corpus. The PoS tagger used in GATE was produced by Hepple (2000), which is a modified version of the Brill tagger (Brill, 1992), and produces a PoS tagger as an annotation for each word or symbol.

4.2.4 Annotations

One of the key features in GATE is that it gives the opportunity to represent information about the text. However, the previous processing resources such as tokeniser and PoS

tagger are running over the text, hence allowing users to obtain various information about the texts being processed. For instance when the tokeniser is applied the outcomes will be words, punctuations, full stop, etc. and when the PoS tagger is applied the outcome will be the sets of noun, verb, pronoun, adjective, etc. as shown in Figure 4.5. However, the user or the developer cannot distinguish between these outcomes unless they are represented by the annotations set.

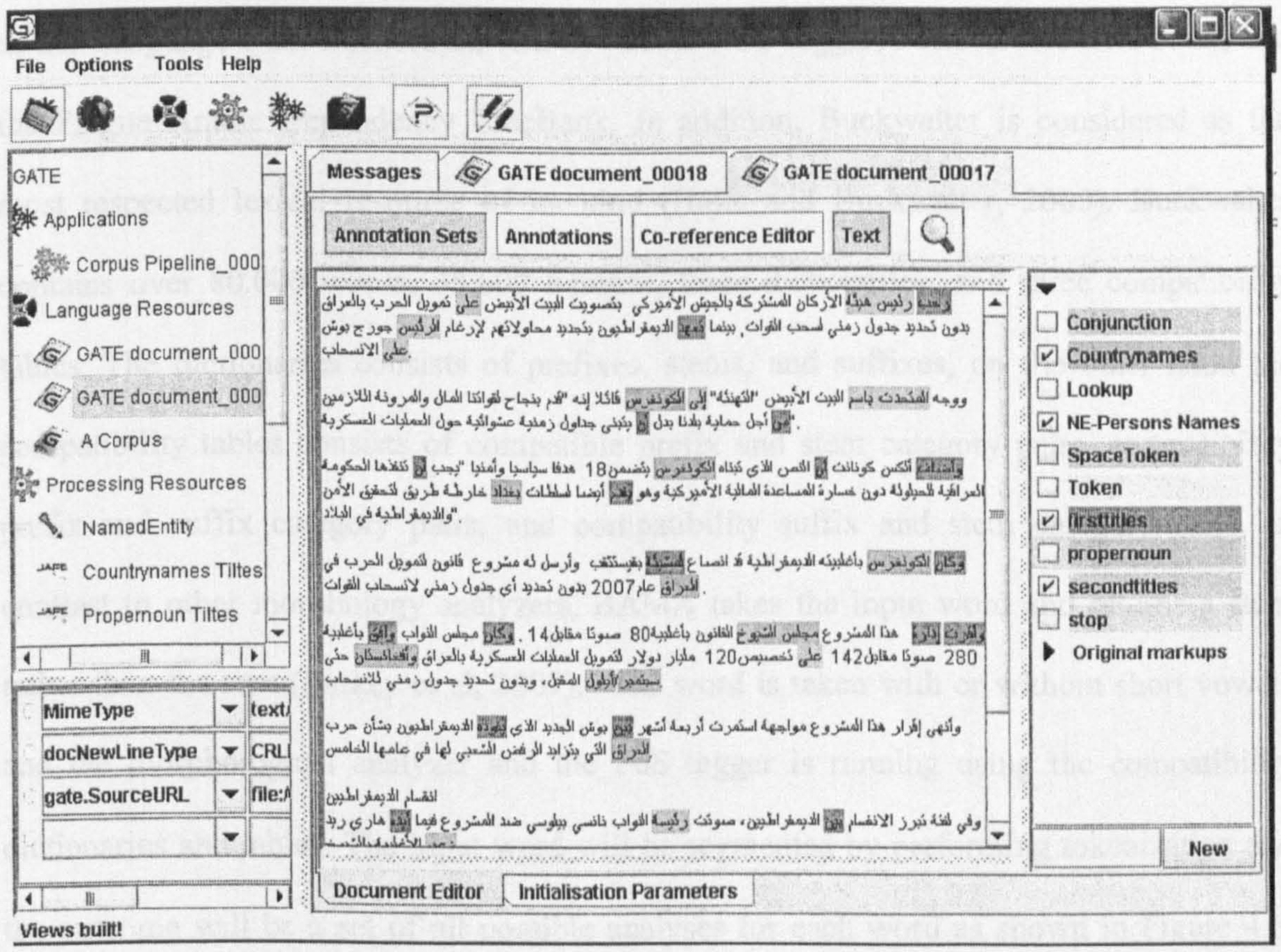

Figure 04.5: Various annotations of an Arabic text.

4.2.5 Language Resources

Language Resources (LRs) are data components such as lexicons, corpora, dictionaries, documents, etc. LR is an essential tool for the developers to create their own applications. Assume that a developer is writing a program that may have a requirement for synonyms; a

number of sources for synonyms are available within GATE such as the Thesaurus. On the other hand to run a named entity program to extract specific information from the text, it is crucial to have a corpus on hand to acquire this purpose. In order to reuse these sources the developer needs to access these data (LRs) from their program.

4.2 Buckwalter Arabic Morphological Analyzer

Buckwalter morphology analyzer (BAMA) is spread and widely used in the literature; such as the Language Data Consortium (LDC) Arabic POS tagger, Peen Arabic TreeBank, and the Prague Arabic Dependency TreeBank. In addition, Buckwalter is considered as the most respected lexical resource of its kind (Hajic and Buckwalter, 2005). Buckwalter contains over 80.000 words, 38.600 lemmas, three dictionaries, and three compatibility tables. The dictionaries consists of prefixes, stems, and suffixes, on the other hand the compatibility tables consists of compatible prefix and stem category pairs, compatibility prefix and suffix category pairs, and compatibility suffix and stem category pairs. In contrast to other morphology analyzers, BAMA takes the input word and returns a stem rather than the root (Larkey et al, 2007). The word is taken with or without short vowels and the morphological analyzer and the PoS tagger is running using the compatibility dictionaries and tables. The input word will be segmented by performing tokenization and the outcome will be a set of all possible analyses for each word as shown in Figure 4.6. BAMA has been made available through the Language Data Consortium (2002).

```
TestBuckwalter.java    NamedEntity.java
        // TODO Auto-generated method stub

        TestBuckwalter tbw = new TestBuckwalter();
        tbw.initializeAraMorph();
        tbw.buckwalterOption("كاتب");

Problems  Javadoc  Declaration  Console
<terminated> TestBuckwalter [Java Application] C:\Program Files\Java\jre1.5.0_05\bin\javaw.exe (Ja
Initializing in-memory dictionary handler...
Loading dictionary : dictPrefixes .
78 entries totalizing 299 forms
Loading dictionary : dictStems ..............................
38600 lemmas and 47261 entries totalizing 82158 forms
Loading dictionary : dictSuffixes ..
206 entries totalizing 618 forms
Loading compatibility table : tableAB ...
1648 entries
Loading compatibility table : tableAC .
598 entries
Loading compatibility table : tableBC ..
1285 entries
... done.
Initializing in-memory solutions handler...
... done.
كاتب = NOUN
كاتب = ADJ
كاتب = VERB_PERFECT
كاتب = NOUN
```

Figure 4.6: Buckwalter output for the word (Kateb, كاتب).

4.3 The Architecture of the System

In the current research, we propose an Arabic IE solution that makes full use of the GATE system and then develop an Arabic IE plug-in to process Arabic information resource based on GATE framework. In addition, our system also uses BAMA, which has been integrated with GATE to make use of its capabilities e.g. processing resource such as Tokeniser, gazetteers, annotations set, etc, and language resources such as, corpora and documents . The architecture of the system is illustrated in Figure 4.7 and its various components described in the following subsections.

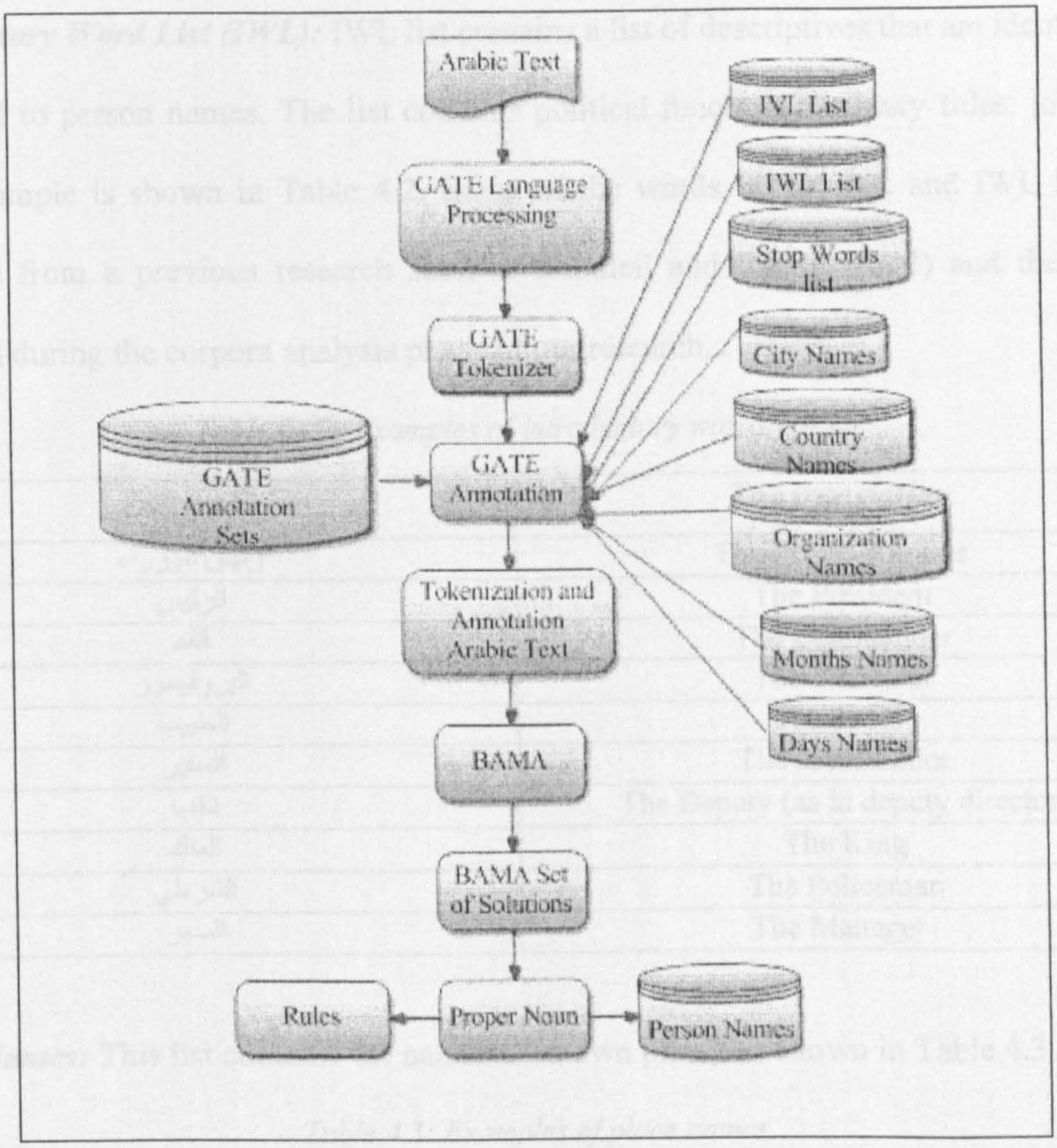

Figure 04.7: The system architecture and the flowchart

Introductory Verb List (IVL): IVL list contains special verbs that are identified as introducing person names. Table 4.1 shows some examples of these verbs.

Table 4.1: Examples of introductory verbs

The word	***The translation***
قال	Say
خرج	Exit
أكد	Assert
أقر	Approve
أعلن	Announce
استقال	Resign
وصل	Arrive
وقع	Sign
غادر	Depart
استقبل	Welcome

Introductory Word List (IWL): IWL list contains a list of descriptives that are identified to be linked to person names. The list contains political functions, military titles, job titles, etc. A sample is shown in Table 4.2. Some of the words in the IVL and IWL lists are collected from a previous research such as (Abuleil and Evens, 2002) and the rest is collected during the corpora analysis phase of our research. .

Table 04.2: Examples of introductory words (IWL)

The word	***The translation***
رئيس الوزراء	The Prime Minister
الرئيس	The President
قائد	The Commander
البروفيسور	The Professor
الطبيب	The doctor
السفير	The ambassador
نائب	The Deputy (as in deputy director)
الملك	The King
الشرطي	The Policeman
المدير	The Manager

Place Names: This list contains the names of known places as shown in Table 4.3.

Table 4.3: Examples of place names

The word	***The translation***
نهر النيل	Nile River
البحر الأبيض المتوسط	Mediterranean Sea
المحيط الأطلسي	Atlantic Ocean
جبال الهمالايا	Himalaya Mountains
الأهرام	Al-Ahram(The Pyramids)
تاج محل	Taj Mahal

Stop Words: This list is the usual list of words that are not important to the application and include prepositions, pronouns, demonstrative, etc as shown in Table 4.4.

Table 04.4: Examples of stop words.

Categories	The Word	The translation
demonstrative nouns	هَذا	Singular masculine: this
	هَذانِ	Dual masculine : these
	هَؤلاءِ	Plural : these
	ذٰلِكَ	Singular masculine: that
	هَذِهِ	Singular feminine : this
relative pronoun	الذي	Singular masculine: who, which
	التي	Singular feminine: who, which
	الذينَ	Plural masculine: who, whom
interrogative particles	كَيْفَ	How
	مَتى	When
	لِماذا	Why
adverbs	هُنا	here
	هُناكَ	there
	تحت	down
incomplete verbs	إنَّ	certainly
	لَعَلَّ	perhaps
	لَكِنَّ	but

Conjunction list: This is the list of conjunctions and Table 4.5 shows some examples.

Table 0.5: Example of conjunctions

The word	*The translation*
من	From
إلى	To
في	In

City and Country Names: This list contains the names of known cities and countries. A sample is given in Table 4.6.

Table 4.6: Example of country names

The word	***The translation***
طرابلس	Tripoli
القاهرة	Cairo
لندن	London
فرنسا	France
إيطاليا	Italy
ألمانيا	Germany
أمريكا	America
الكويت	Kuwait
اليابان	Japan
بغداد	Baghdad

Organisation Names: This list contains the names of known organisations; Table 4.7 shows some of these organisations.

Table 04.7: Example of organization names

The word	***The translation***
الأمم المتحدة	The United Nations
الكونغرس	The Congress
مجلس الأمن	The Security Council
البيت الأبيض	The White House
البنك المركزي	The Central Bank
شركة مايكروسفت	Microsoft Corporation
الإتحاد الأوروبي	The European Union
البنتاقون	The Pentagon
صحيفة الإنديبندت	The Independent Newspaper
الكرملن	The Kremlin

Arabic Person Names: There are a limited number of Arabic names that start with the letters Alif and Lam (ال, AL). These are very often confused with common names as the letters (ال, AL) are the equivalent of the English definite article (the). As these names are known and limited, the efficiency and precision of the system were largely improved by manually developing a list of all known Arabic names that starts with AL. A sample is given in Table 4.8. We gathered around three hundred names from this type of names.

Table 4.8: Examples of Arabic person names starting with "AAL"

The word	***The transliteration***
المهدي	aalmhdy
الصادق	aalsaadq
الحرمين	aalhrmyn
الطاهر	aaltahr
الطيب	aaltyb
المالكي	aalmaalky

All these components (GATE, BAMA and the various lists) are used in our application. The raw Arabic text is taken from the collection of documents (corpora) via the Language Resources (LRs) component then the entire text is tokenized using the GATE tokeniser component which outputs are tokens representing words, numbers, symbols and spaces. Once the tokenization process is completed, the system will read the gazetteers containing all the lists mentioned above and the identified tokens from the texts are highlighted according to which list they belong to. Subsequently, BAMA will be used to identify the various PoS such as verbs, nouns and adjectives. The final step of the process is the application of the rules developed to obtain Arabic Named Entity. The rules will be discussed in details in chapter 5.

4.4 Summary

In this chapter we introduced the overall architecture of our system and described its components. Our architecture made use of very well known tools in the field of natural language processing mainly GATE and BAMA. These two systems were described in details in addition to the various lists we identified as part of our architecture.

Chapter 5

The Development of the Rules

5.1 Introduction

As we have mentioned it in chapter 4, our application relies on various sources such as Language Resources corpora and documents and Processing Resources such as tokeniser, gazetteer and POS tagger. In addition BAMA was used alongside these resources. However, to achieve the main objective of this research, which is identification of Named Entities in Arabic, we developed a set of rules specific to the Arabic language. We adopted a Rule Based system instead of Automatic Training systems (statistical) for the following reasons:

- We do not have access to a large annotated Arabic corpora and developing one will require a long time that may hinder the development of this research. This is a requirement for statistical systems.
- Rule based systems can work well with small training corpora.
- Rule based systems are useful for limited domains such as our domain which is the political domain.
- Rule based systems can be used with both well-formed and ill-formed input whereas with statistical approaches it is not easy to work with ill-formed input.
- When using statistical systems, some changes may require re-annotation of the entire training corpus
- Given the morphological structure of the Arabic language, we believe the rule based approach is more appropriate to the nature of the Arabic language.

In MUC-6, three subtasks have been identified for the NER task. These are:

i. TIMEX: temporal expression (time and date)

ii. NUMEX: numerical expression of percentage, monetary expressions, etc.

iii. ENAMEX (for the proper names), was defined as the extraction of proper names and classification of each one of them as, organization (named corporate, governmental, or other organizational entity); location (name of politically or geographically defined location) or person (named person or family).

5.2 ENAMEX: Proper Name Recognition

This sub-task is the one we are concerned with in general and the person names in particular. Person names in the Arabic language are very difficult to recognize and extracting from the text as the proper names do not start with a capital letter as is the case in many other languages. This represents a considerable difficulty for Arabic NER. In addition, there are hardly clues in the Arabic text that can help with the identification of proper names. The identification of other named entities is less problematic and there are usually clues that can help in their identification. For examples words such as company, Kingdom, Republic and Authority may indicate the presence of a named entity as the examples in Table 5.1 illustrate. Hence, more efforts are concentrated on persons names as they are considered to be the most challenging for the Arabic language (Shaalan and Raza, 2007). Hence, in the case of organisation and location names our system will rely on these keywords (i.e. trigger words) and the defined gazetteers.

Table 5.1: Examples of clues associated with some named entities

The phrase	*The translation*	*The clue (trigger word)*	*The named entity type*
هيئة الأمم المتحدة	United Nation Authority	(هيئة, Authority)	organisation
منظمة حقوق الإنسان	Human Rights Organisation	(منظمة, Organisation)	organisation
جمهورية مصر	Arab Republic of Egypt	(جمهورية, Republic)	location
المملكة المتحدة	United Kingdom	(المملكة, Kingdom)	location
شركة الغاز	Gas company	(شركة, company)	organisation

5.2.1 Arabic Person Name Recognition

The arrangements of Arabic statements do not have a fixed format but depend mainly on the writing styles. Proper names may appear in any position within the sentence and can be next to a keyword of far away from it. Hence we have identified a set of rules to deal with all these cases. An example of these structures is given in Table 5.2. In this example, the keyword is (رئيس, President) and the proper noun is (معمر القذافي, Muammar Al- Gadhafi). The proper name is six words away from the keyword.

Table 5.2: An example showing the position of a proper name with regards to its keyword.

The Arabic Sentence	*The English translation*
رئيس الجماهيرية العربية الليبية الشعبية الإشتراكية العظمى معمر القذافي	President of Socialist People's Libyan Arab Great Jamahiriya "state of the masses" Muammar Al- Gadhafi.

The development of the Introductory Verb List (IVL) and the Introductory Word List (IWL) play a central role in the development of the heuristics and are added to the GATE system. We note here that the words in IVL and IWL are not candidate person names but are only used as keywords to find the position of person names in the text. The heuristics are based on the position of IVL and IWL words in the text and other words around them.

5.2.1.1 Proper Noun Preceding a Keyword

In this section we will present the first proper noun rule which deals with proper nouns appearing before a keyword. The rule is presented in Figure 5.1.

Read word *w* from the text	
IF	*w* belongs to IWL or IVL
THEN	IF previous word next to *w* belongs to Stop Words
	THEN find in the text the next word belonging to IWL or IVL
	ELSE process word by BAMA

Figure 5.1: The first proper noun rule

This rule is illustrated by the example given in Table 5.3

Table 5.3: An example of a proper noun preceding a keyword

The Arabic Sentence	***The English translation***
عبد الله بن سعود الملك السعودي سيغادر من الرياض إلى لندن	Abdullah bin Saud the Saudi king will travel from Riyadh to London

In this example the keyword is (الملك, the king) and belongs to IWL and the proper noun (عبد الله بن سعود, Abdullah bin Saud) is preceding the keyword. As we note in this example, the word preceding the keyword is not a stop word, in this case the word should be taken and tested by BAMA to check whether the word preceding the keyword is proper noun or not. Sometimes the word preceding the keyword is a stop word as in the example given in Table 5.4. In this case our system ignores the current keyword and looks for the next one. Where the word (ضابط, officer) is the keyword belonging to IWL and the word (مع, with) is the stop word.

Table 5.4: An example showing a stop word preceding a keyword

The Arabic Sentence	***The English translation***
جاء السائق مع ضابط الشرطة	The driver came with the police officer

5.2.1.2 Proper Noun Located Next to the Keyword

The second proper noun rule is developed to deal with proper nouns appearing next to a keyword, with other words such as stop words; IVL, IWL being next to the keyword instead of the proper noun. Therefore, we need to examine the word next to the keyword before we mark it as a proper noun. Obviously in this case a particular procedure is required to resolve this case:

- First we will present the second proper noun rule when the word next to the keyword is a stop word as in Figure 5.2

```
Read word w from the text

IF      w belongs to IVL or IWL
THEN    IF next word belongs to Stop Words
            THEN find in the text the next word
                    belonging to IVL or IWL
```

Figure 05.2: Proper noun rule when a stop word is next to a keyword

This rule is illustrated by the example given in Table 5.5

Table 05: Example showing a stop word next to a keyword

The Arabic Sentence	***The English translation***
أعلن في المؤتمر الأول	Announced in the first conference

Where the word (في, in) belongs to the stop word list. Thus the system will move from the current keyword to the next keyword in the text.

- Second we will present the second proper noun rule when the word next to the keyword belongs to IVL as shown in Figure 5.3

```
IF      w belongs to IVL or IWL
THEN  IF next word belongs to IVL
          THEN ignore the first word and use the second as a starting point
          ELSE process word by BAMA
```

Figure 5.3: The proper noun rule when an IVL word is next to a keyword

The example in Table 5.6 illustrates this rule. However the keyword in this example belongs to IVL (قال, said) and the word next to the keyword belongs to IVL as well (السيد, Mr.) in this case the first keyword(قال, said) will be ignored and the next keyword (السيد, Mr.) will be used as a keyword.

Table 5.6: An example showing an IVL word next to a keyword

The Arabic Sentence	***The English translation***
قال السيد محمد خالد	Mr. Muhammad Khaled said.

- Third, sometimes the word next to the keyword belongs to the IWL; hence in this case the second proper noun rule shown in Figure 5.4 will be applied.

```
IF      w belongs to IVL or IWL
THEN  IF next word belongs to IWL
          THEN ignore the first word and use the second as a starting point
          ELSE process word by BAMA
```

Figure 05.4: The proper noun rule when an IWL word is next to a keyword

The example given in table 5.7 illustrates this rule, where the word (غادر, departure) is the keyword and belongs to IVL and the word (الرئيس, the president) belongs to IWL. Therefore the system will move from the current keyword to the next keyword.

Table 5.7: An example showing an IVL word next to a keyword

The Arabic Sentence	***The English translation***
غادر الرئيس السوري من المطار	The Syrian president departed from the airport

- Finally, when the word next to the keyword is not a member of any list (gazetteer) or any of the previous type of words i.e. stop word, IVL, IWL then the word is most likely to be a proper noun. However, to verify whether this word is a proper noun or not we need to use BAMA and we will discuss this issue in section 5.3.

5.2.1.3 The Proper Noun is Located away from the Keyword

As we have mentioned above, the proper noun in Arabic sentences can be positioned far from the keyword as in the example given in Table 5.8. However, through the experiments we have conducted over a huge amount of articles, we observed that the words between the trigger word (keyword) and the assumed proper noun in Arabic sentences are adjectives agglutinant from the beginning by a definite article (ال, the) as shown in the example given in Table 5.8.

Table 05.8: An example illustrating an adjective between a keyword and a proper noun

The Arabic Sentence	***The English translation***
رئيس الوزراء البريطاني السابق توني بلير	The previous British Prime Minister Tony Blair

Where the word (رئيس, Prime) is the keyword and the words (توني بلير, Tony Blair) represent a proper noun and all the words between the keyword and the proper noun are

adjectives. However, in some cases the words between the keyword and the proper noun are not adjectives. During the analysis of our corpora we observed that this occurs in two cases. The first case is when one of these words is a proper noun starting with the definite article (ال, the) as in the example given in Table 5.9. The second case is when one of these words is a conjunction followed by a country or city name as in the example given in Table 5.10. However we generated the third proper noun rule given in Figure 5.5 to solve these situations.

```
IF      w belongs to IWL or IVL
THEN    IF next word starts with AL (alif and lam)
        THEN PROCESS_AL_WORDS (w)
                WHILE w starts with AL
                 IF w belongs to list Arabic_Person_Names
                 THEN select w as Person name
                 ELSE w = next word in text
                 PROCESS_AL_WORDS (w)
                IF w is a conjunction and the next word belongs to Country
                   or Place lists
                THEN {ignore the conjunction and the next word

                process the next word by BAMA}
```

Figure 05.5: Proper noun rules for nouns starting with (ال, the) or when a conjunction is followed by a country or city name

The example given in table 5.9 illustrates the third proper noun rule, where the keyword in this example (الملك, the king) belongs to IWL and the words (المغربي, the Moroccan's) and (السابق, the previous) are adjectives starting with the definite article (ال, the) and both these words do not belong to the Arabic Person Names list which we have introduced in section 4.3. Consequently both words will be ignored and the word (الحسن, Hassan) will be selected as a person name since it is in the Arabic Person Names list.

Table 05.9: An example showing a proper noun starting with definite article (ال)

The Arabic Sentence	***The English translation***
قال الملك المغربي السابق الحسن الثاني	The previous king of Morocco Hassan II said

On the other hand the example given in table 5.10 illustrates the rule when the conjunction followed by the city name, where the word (السفير, Ambassador) belongs to IWL and the word (الامريكي, American) is an adjective starting with the definite article (ال, the) and not in the Arabic Person Names list. Hence the system will ignore the word (الامريكي, American) and the word (في, in) which is a conjunction. The next word (بغداد Baghdad) is in the City and Country lists., Hence the both words will ignored and the next word which (زلماي, Zalmay) will be submitted to BAMA.

Table 05.10: An example showing a conjunction followed by the city name.

The Arabic Sentence	***The English translation***
السفير الامريكي في بغداد زلماي خليلزاد	The American Ambassador in Baghdad Zalmay Khalilzad

5.2.1.4 The Morphological Analysis Stage

As we mentioned in chapter 3, morphological analysis and a PoS tagger are considered as essential tools in NLP in general and NE in particular. Furthermore, these tools are crucial whether a rule based or an automatic training system (statistical) is used. In addition the Arabic language is a language with a very complex morphology; each word in the text has more than one morphological analysis. Thus overcoming ambiguity is the major challenge for NLP in Arabic (Kamir et al, 2002). Consequently additional attention must be paid to these essential tools to obtain precise results. BAMA, one of the best known, well

documented, morphological analyzers for Modern Standard Arabic (MSA) (Attia, 2006) is used to disambiguate words whenever the extraction rule is not confident about the category of a specific word that can be a proper noun. The rule shown in Figure 5.6 is then used.

```
IF among the words returned by BAMA there is a word w that is a proper name
THEN IF w is not in Countries, Places and Organizations Lists
        THEN SELECT w as a person name;
        ELSE ignore w
ELSE IF no solution is provided by the BAMA system
        THEN select word as person name
```

Figure 5.6: Rule for selecting a proper name from BAMA

The example given in Table 5.11 illustrates this rule, where the word (رئيس, prime) is the keyword belonging to IWL and the words (الوزراء, minister) and (العراقي, the Iraqi) are adjectives starting with a definite article (ال, the) therefore both must be ignored.

Table 5.11: An example showing an adjective followed by a proper noun.

The Arabic Sentence	***The English translation***
قال رئيس الوزراء العراقي نوري المالكي	The Iraqi prime minister Nouri Al-Maliki said

The next word (نوري, Nouri) tested by BAMA and the possible outputs for this word are shown in Figure 5.7. Obviously, one of these results is a proper noun; consequently our system will perform the last stage to verify whether this proper noun is a person name or any other proper noun such as organization, location, etc. However, our system will also check if the word (نوري, Nouri) is in the City and Country names list. In this case it is not a member of this list, hence the word is extracted as a person name.

```
Initializing in-memory dictionary handler...
Loading dictionary : dictPrefixes .
78 entries totalizing 299 forms
Loading dictionary : dictStems .....................
38600 lemmas and 47261 entries totalizing 82158 forms
Loading dictionary : dictSuffixes ..
206 entries totalizing 618 forms
Loading compatibility table : tableAB ...
1648 entries
Loading compatibility table : tableAC .
598 entries
Loading compatibility table : tableBC ..
1285 entries
... done.
Initializing in-memory solutions handler...
... done.
نوري = NOUN
نوري = ADJ
نوري = VERB_IMPERFECT
نوري = NOUN
نوري = VERB_IMPERFECT
نوري = NOUN
نورى = NOUN PROP
```

Figure 05.7: Buckwalter output for the word (نوري Nouri)

Sometimes among the results returned by BAMA is a proper noun but not necessarily a person name as the example given in Table 5.12.

Table 05.12: An example showing a place named entity.

The Arabic Sentence	***The English translation***
أعلنت دبي عن الفائزين	Dubai announced the winners

Where the word (أعلنت, announced) is the keyword and belongs to the IVL list although the word (دبي, Dubai) is a proper noun as shown in Figure 5.8, but our system will not mark this word as a person name, because the word (دبي, Dubai) belongs to the City and Country names list.

```
Initializing in-memory dictionary handler.
Loading dictionary : dictPrefixes .
78 entries totalizing 299 forms
Loading dictionary : dictStems .....
38600 lemmas and 47261 entries totalizing
82158 forms
Loading dictionary : dictSuffixes ..
206 entries totalizing 618 forms
Loading compatibility table : tableAB .
1648 entries
Loading compatibility table : tableAC .
598 entries
Loading compatibility table : tableBC .
1285 entries
... done.
Initializing in-memory solutions handler.
... done.
possible analysis of the input word دبي =NOUN
possible analysis of the input word دبي =ADJ
possible analysis of the input word دبي =NOUN
possible analysis of the input word دبي =NOUN_PROP
```

Figure 05.8: Buckwalter output for the word (دبي Dubai)

However there are cases where BAMA does not recognise a particular word and will not provide a solution to it as shown in Figure 5.8. We noticed that in all the cases we have seen so far, this usually points to a non Arabic Proper Name as illustrated in the example given in Table 5.13.

Table 05.13: An example where the proper noun is missed by BAMA.

The Arabic Sentence	***The English translation***
رئيس فريق المفاوضات الإيرانية لاريجاني	The team leader of Iranian negotiation Larigany

Where the sequence of the words (رئيس فريق) are a keyword belonging to IWL and both words (المفاوضات, negotiation) and (الإيرانية, Iranian) are adjectives starting with a definite article (ال, the). Therefore they should be ignored and the word (لاريجاني, Larigany) will be processed by BAMA. Although the system did not provide any solution, the word is selected as a person name.

```
Initializing in-memory dictionary handler.
Loading dictionary : dictPrefixes .
78 entries totalizing 299 forms
Loading dictionary : dictStems .....
38600 lemmas and 47261 entries totalizing
82158 forms
Loading dictionary : dictSuffixes ..
206 entries totalizing 618 forms
Loading compatibility table : tableAB ...
1648 entries
Loading compatibility table : tableAC .
598 entries
Loading compatibility table : tableBC ..
1285 entries
... done.
Initializing in-memory solutions handler...
... done.
possible analysis of the input word لارجاني =  No Solution
```

Figure 05.9: Buckwalter output for the word (لارجاني, Larigany)

5.2.2 Organisation and Location Recognition

As we mentioned early the most challenges for the task Arabic NER is the recognizing and extracting the person name from the text. With respect to the organisation we have gathered two hundred and sixty seven various organisations names. However, the organisation names in Arabic are very often written in different forms depending on different Arab regions or countries. Examples of these differences are given Table 5.14. On the other hand, with regards to the locations, we collected around five thousand entries from several sources e.g. GATE, internet, Wikipedia, etc. These entries combined countries, cities, rivers, mountains etc and these are organised in different gazetteers that are used by our system.

Table 05.14: illustrate the variation of the name of the organisation in Arabic text

The organisation	***The organisation written in Arabic***
BBC	هيئة الإذاعة البريطانية
	البي بي سي
	بي بي سي
	محطة الإذاعة البريطانية
Congress	الكونقرس
	الكونغرس
Microsoft	المايكروسفت
	مايكروسفت
	الميكروسفت
	ميكروسفت
	مايكروسوفت
	ميكروسوفت
CIA	وكالة المخابرات الأمريكية
	وكالة الإستخباترات الأمريكية
	وكالة المخابرات المركزية
	السي أي إيه
	سي أي إيه

The organisation and location names are recognized mainly in the morphological analysis stage by looking up in the lists of organisation, cities, states, countries, and other place names. Similar to the organisations names, the location in Arabic text are also written in various form as the examples in Table 5.15 show. 5.3 TIMEX: Date and Time Recognition.TIMEX phrases are temporal expressions, which are subdivided into date expressions (٢٦ أيار ٢٠٠٩, 26 May 2009) and time expressions (السادسة مساء, 6:00 pm). However we conducted a survey of several official Arabic newspapers and we noticed that the Arab countries do not use a unique form for the months of the year, there are variations in names and also spelling.

Table 5.15: An example of name variation in Arabic location names.

The location	***The location written in Arabic***
USA	أمريكيا
	أمريكا
	الولايات المتحدة
	الولايات المتحدة الأمريكية
NATO	الناتو
	الحلف الأطلسي
	حلف الناتو
Mediterranean Sea	البحر الأبيض
	البحر المتوسط
	البحر الأبيض المتوسط
Nile river	النيل
	نهر النيل

We split Arabic months based on their use in different Arab countries into seven groups as shown in Table 5.16.

Table 5.16: The different months' names as used by different countries.

The Romanian months	*The gulf countries, Yemen, and Egypt*	*Iraq, Syria, Lebanon, Jordan, and Palestine*	*Algeria and Tunisia*	*Libya*	*Morocco*	*Mauritania*	*The Arabic months*
January	يناير	كانون الثاني	جانفي	أي النار	يناير	يناير	محرم
February	فبراير	شباط	فيفري	النوار	فبراير	فبراير	صفر
March	مارس	آذار	مارس	الربيع	مارس	مارس	ربيع الأول
April	أبريل	نيسان	أفريل	الطير	أبريل	إبريل	ربيع الثاني
May	مايو	أيار	ماي	الماء	ماي	مايو	جمادى
June	يونيو	حزيران	جوان	الصيف	يونيو	يونيو	جمادى
July	يوليو	تموز	جويلية	ناصر	يوليوز	يوليو	رجب
August	أغسطس	آب	أوت	هانيبال	غشت	أغشت	شعبان
September	سبتمبر	أيلول	سبتمبر	الفاتح	شتنبر	شتمبر	رمضان
October	أكتوبر	تشرين الأول	أكتوبر	التمور	أكتوبر	أكتوبر	شوال
November	نوفمبر	تشرين الثاني	نوفمبر	الحرث	نوفنبر	نوفمبر	ذو القعدة
December	ديسمبر	كانون الأول	ديسمبر	الكانون	دجنبر	دجمبر	ذو الحجة

Contrary to the month's names, All Arab countries use the same names for the days of the week as shown in Table 5.17. Two gazetteers for the Months and Days have been developed.

Table 05.17: The days of the week in the Arabic language

The Arabic day	*The English translation*
السبت	Saturday
الأحد	Sunday
الإثنين	Monday
الثلاثاء	Tuesday
الإربعاء	Wednesday
الخميس	Thursday
الجمعة	Friday

5.4 NUMEX: Percent and Money Expression Recognition

NUMEX phrases are numeric expressions, which are subdivided into percent expressions (٪ ٣.٤, 3.4%) and money expressions (٢٤٥ مليون باوند, £245 million). However this type of

entity is considered as uncomplicated NE, since it can be recognised straightforward from the text. In the Arabic language, as in other languages, the numbers are composed of digits or the written names of the numbers as shown by the examples given in Table 5.18.

Table 05.18: The digital and alphabetic numbers in Arabic

Digital number	***Digital number in Arabic***	***Alphabetic number***	***Alphabetic number in Arabic***
1	١	One	واحد
7	٧	Seven	سبعة
45	٤٥	Forty five	خمسة وأربعون
74	٧٤	Seventy four	أربعة وسبعون
105	١٠٥	One hundred and five	مائة وخمسة
204	٢٠٤	Two hundred and four	مائتين وأربعة
515	٥١٥	Five hundred and fifteen	خمسمئة وخمسة عشر

Eventually, with regard to currencies, we have collected several types of currencies and we locate them in a gazetteer as shown in Table 5.19.

Table 5.19: A sample of currency names in Arabic

The Arabic currencies	***The English translation***
باوند	Pound
دولار	Dollar
يورو	Euro
دينار	Dinar
ريال	Real
ليرة	Lira
شيكل	Shekel
فرنك	franc

5.5 Summary

After investigating the characteristics of the Arabic language and the approaches used in the field of the NE recognition and justified the choice of a rules based system for Arabic NER, in this chapter we developed the rules that allowed our system to identify the

different entities. In this chapter, we noticed that people names are the most difficult to identify. For the other entities, developing gazetteers and using them for the identification of other entities are more efficient and appropriate. BAMA was used to clarify the classification of a proper name as a person's name.

Chapter 6

System Implementation and Design

6.1 Introduction

In this chapter we will show how our application is designed and implemented. Moreover as we illustrated in chapter 4 our application combined several components including GATE, BAMA, the different gazetteers and a set of rules which have been implemented. In this chapter we describe how these programs operate and are used by our system.

6.2 System Design

In this section we will give a brief description of the class diagram of our system as shown in Figure 6.1. The main class in our system is the Arabic Named Entity Finder (ANEFinder) class which contains the main function. The GATE system is also initialized and called from this class as shown in Figure 6.2. This class is related to other classes of the system that are: NamedEntity, ManipulateCorpus and ManipulateJapeRules classes.

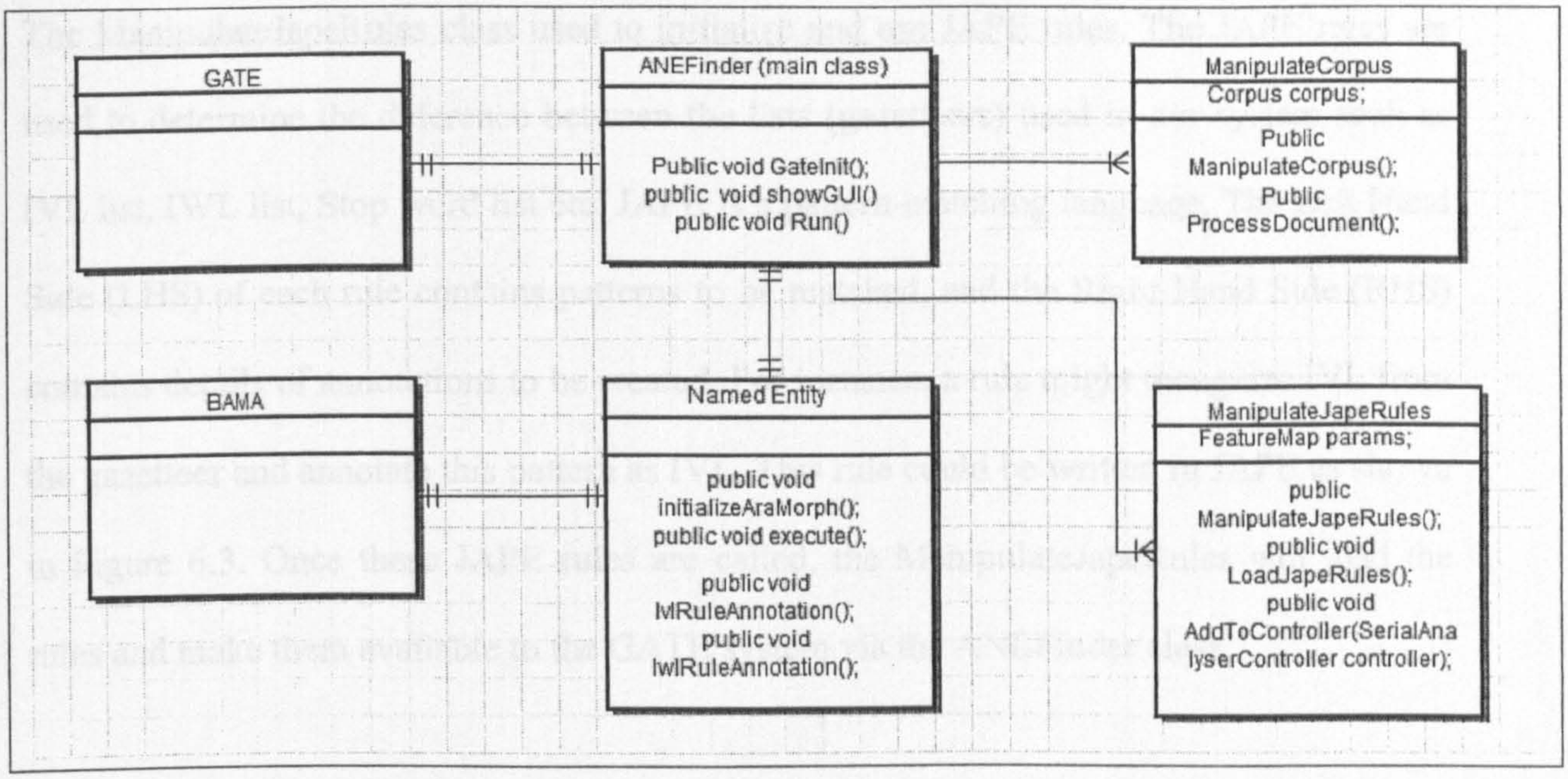

Figure 6.1: class diagram for our system.

The ManipulateCorpus class initializes the corpus used in the GATE system. All documents used in our application are loaded, processed and prepared in this class. These documents are then provided by ANEFinder to the GATE via the language resources (LR).

```
public static void main(String[] args) {
        // TODO Auto-generated method stub
        try
        {
                Gate.setGateHome(new File("C:\\Program Files\\Gate-5.0"));
                Gate.setUserConfigFile(new File(WorkingDirectory + "gate.xml"));

                Gate.init();

                showGUI();


        }

        catch (Exception e)
        {
                //e.printStackTrace();
        }

}

public static void showGUI()
{
        MainFrame mainFrame = new MainFrame();
        mainFrame.setSize(800, 600);
        mainFrame.setVisible(true);
}
```

Figure 6.2: initializing GATE within ANEFinder class

The ManipulateJapeRules class used to initialize and use JAPE rules. The JAPE rules are used to determine the deference between the lists (gazetteers) used in our system such as IVL list, IWL list, Stop word list etc. JAPE is a pattern-matching language. The Left Hand Side (LHS) of each rule contains patterns to be matched, and the Right Hand Side (RHS) contains details of annotations to be created. For instance, a rule might recognize IVL from the gazetteer and annotate this pattern as IVL. This rule could be written in JAPE as shown in Figure 6.3. Once these JAPE rules are called, the ManipulateJapeRules will load the rules and make them available to the GATE system via the ANEFinder class.

```
Phase:TitleFinder

Input:Lookup

Rule:TitleConverter
(
{Lookup.majorType == "IVL"}):match
)
-->
:match.IVL = {}
```

Figure 6.3: The JAPE rule for IVL

The final class is NamedEntity where BAMA is initialized and this class contains the actual code written for our rules to manipulate and specify the named entities. Figure 6.4 shows the initialization of BAMA and also shows how the word has been checked by BAMA to confirm whether the proper noun exists within the outputs of this word or not.

```
import gpl.pierrick.brihaye.aramorph.*;
private AraMorph araMorph;

public void initializeAraMorph()
        {
                //initialize arabmoprph

                araMorph = new AraMorph(null, false);
        }

public boolean buckwalterOption(String matchedWord)
        {
                if(araMorph.analyzeToken(matchedword, false))
                {
                        HashSet solutions = araMorph.getWordSolutions(matchedword);

                        Iterator solutionIterator = solutions.iterator();
                        //System.out.println("**********************************");
                                while (solutionIterator.hasNext()   )
                                {
                                        Solution solution = (Solution)solutionIterator.next();

                                String stem = solution.getStemPOS();
                                //System.out.println(matchedword + " = " + stem);
                                if(stem.compareTo("NOUN_PROP")==0)

                                        return true;
                        }
                }
            else
              return true;

            return false;
        }
```

Figure 6.4: initializing BAMA within NamedEntity class and testing the word by BAMA.

6.2 The Implementation Environment

All the programs we used in our application need an appropriate development environment to be implemented. GATE, which is the main component of our system, is implemented using the JAVA programming language, one of the most powerful and popular general purpose programming languages available. Furthermore, BAMA is implemented using several programming languages including JAVA. Consequently, it was therefore essential to seek an environment appropriate to GATE and BAMA and we choose the JAVA programming language to implement our system. There are several JAVA Integrated Development Environments (IDE) freely available such as Eclipse and Netbeans. For the implementation of our system, we choose the Eclipse IDE. Although the Eclipse Platform has a lot of built-in functionality, most of that functionality is very generic. Moreover,

additional tools can be used to extend the platform to work with new content types. The Platform also provides useful building blocks and frameworks that facilitate developing new tools. However the Eclipse Platform is designed and built to meet the following requirements (IBM Corporation, 2006):

- Support the construction of a variety of tools for application development.
- Support an unrestricted set of tool providers, including independent software vendors (ISVs).
- Support tools to manipulate arbitrary content types (e.g., HTML, Java, C, and XML).
- Facilitate seamless integration of tools within and across different content types and tool providers.
- Support both GUI and non-GUI-based application development environments.
- Run on a wide range of operating systems, including Windows and Linux.
- Capitalize on the popularity of the Java programming language for writing tools.

6.3 The System Implementation

In this section we will highlight how our application is implemented, the first step towards building our application is establishing a workspace in eclipse. The workspace is the physical location where we are working in. The entire application will be stored and saved in this workspace. The workspace must be given a specified name and be located in a known place as shown in Figure 6.5.

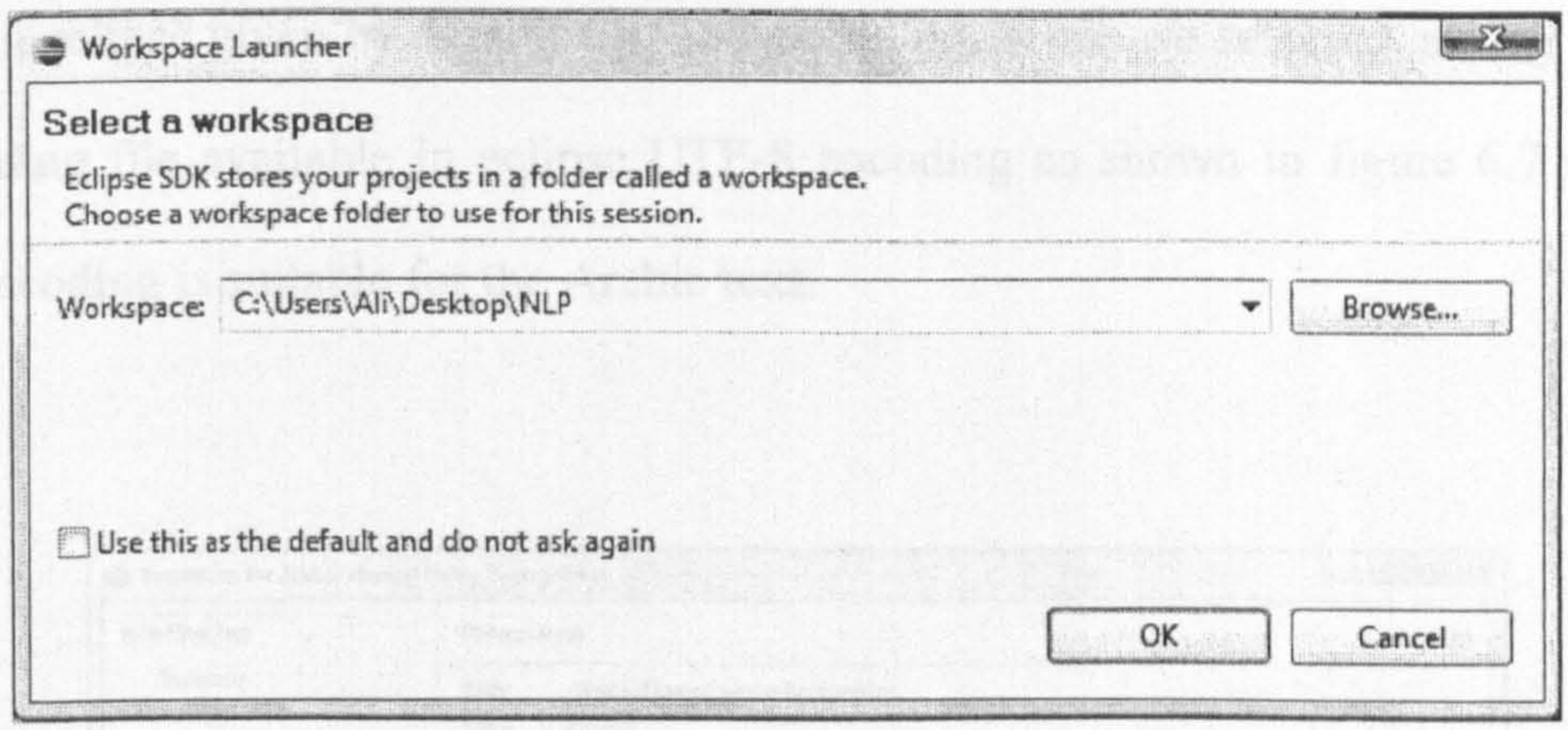

Figure 06.5: The workspace of our application

Following the first step, we need to create our project to gathering our systems e.g. GATE, BAMA, etc. This project named "Arabic Named Entity Recognition" is shown in Figure 6.6. Hence all the operations of our application conducted at this project.

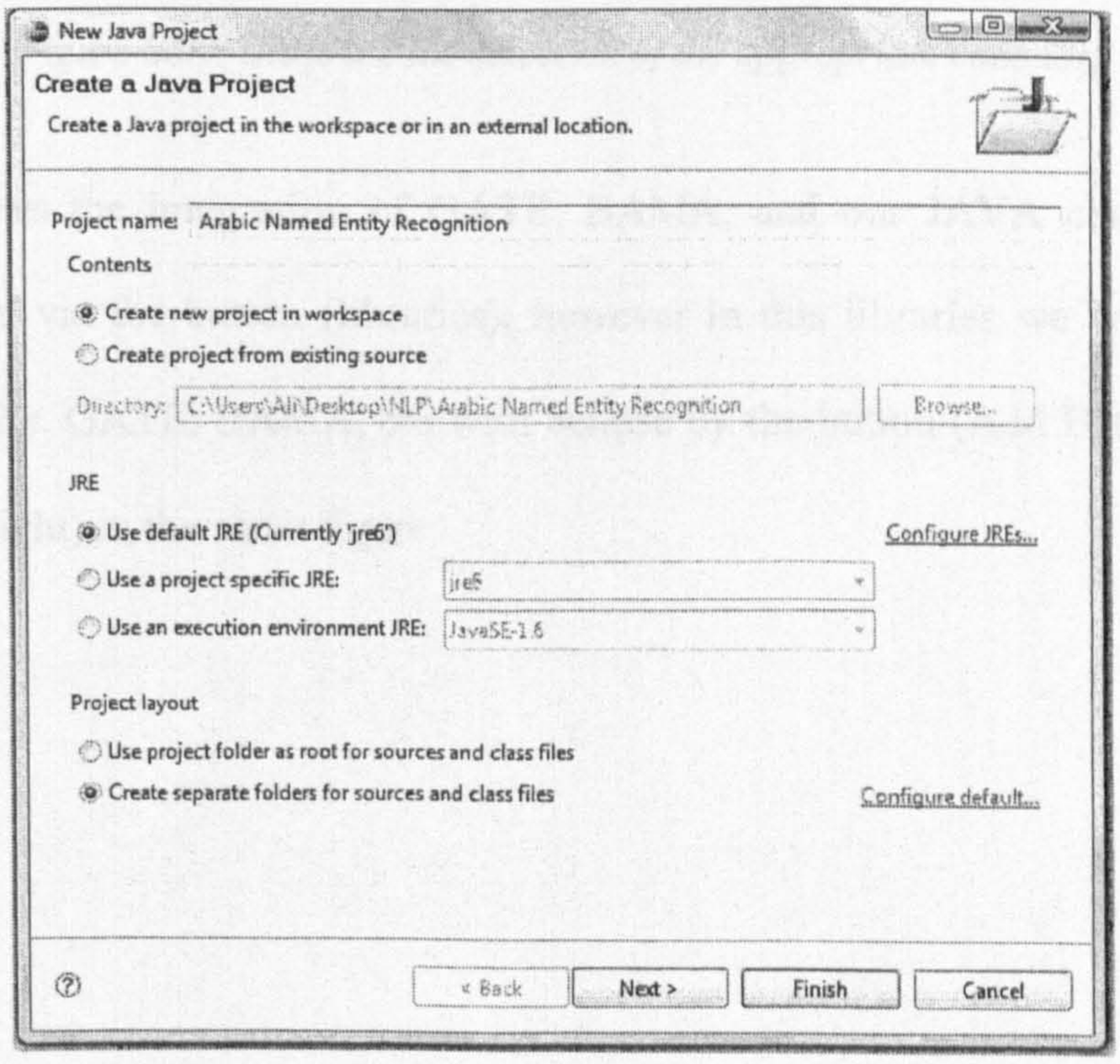

Figure 6.6: illustrate the creation of our project

the text in eclipse given by default Cp1252 encoding, hence we selected among the list of the encoding file available in eclipse UTF-8 encoding as shown in figure 6.7 where the UTF-8 encoding is suitable for the Arabic text.

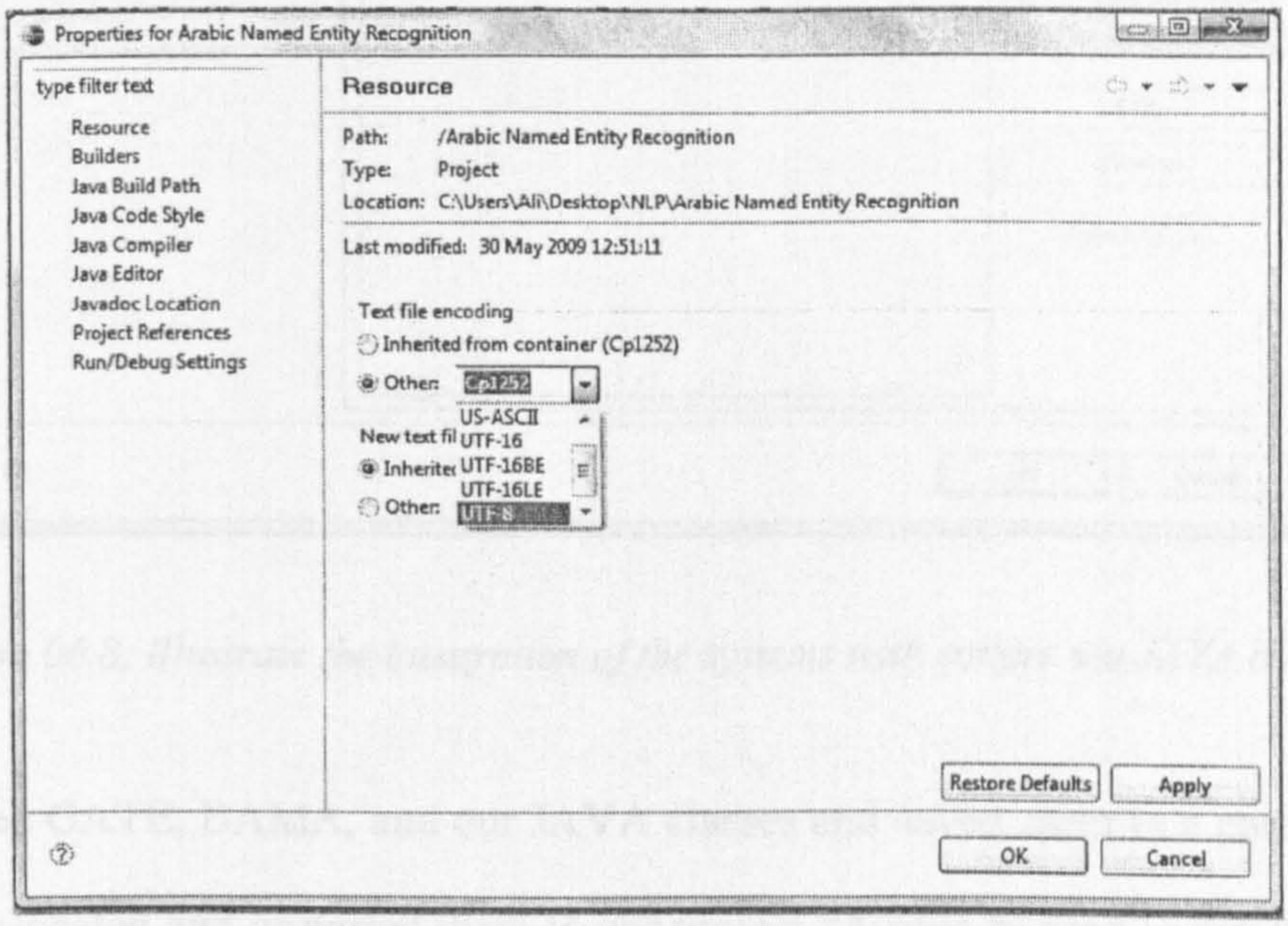

Figure 06.7: illustrate the selection of the appropriate encoding

Figure 6.8 shows the integration of GATE, BAMA, and our JAVA classes within the eclipse platform via the button (libraries), however in this libraries we will integrate the entire systems i.e. GATE, BAMA, etc with eclipse by the button (Add External JARS) as shown in (top right) in the same figure.

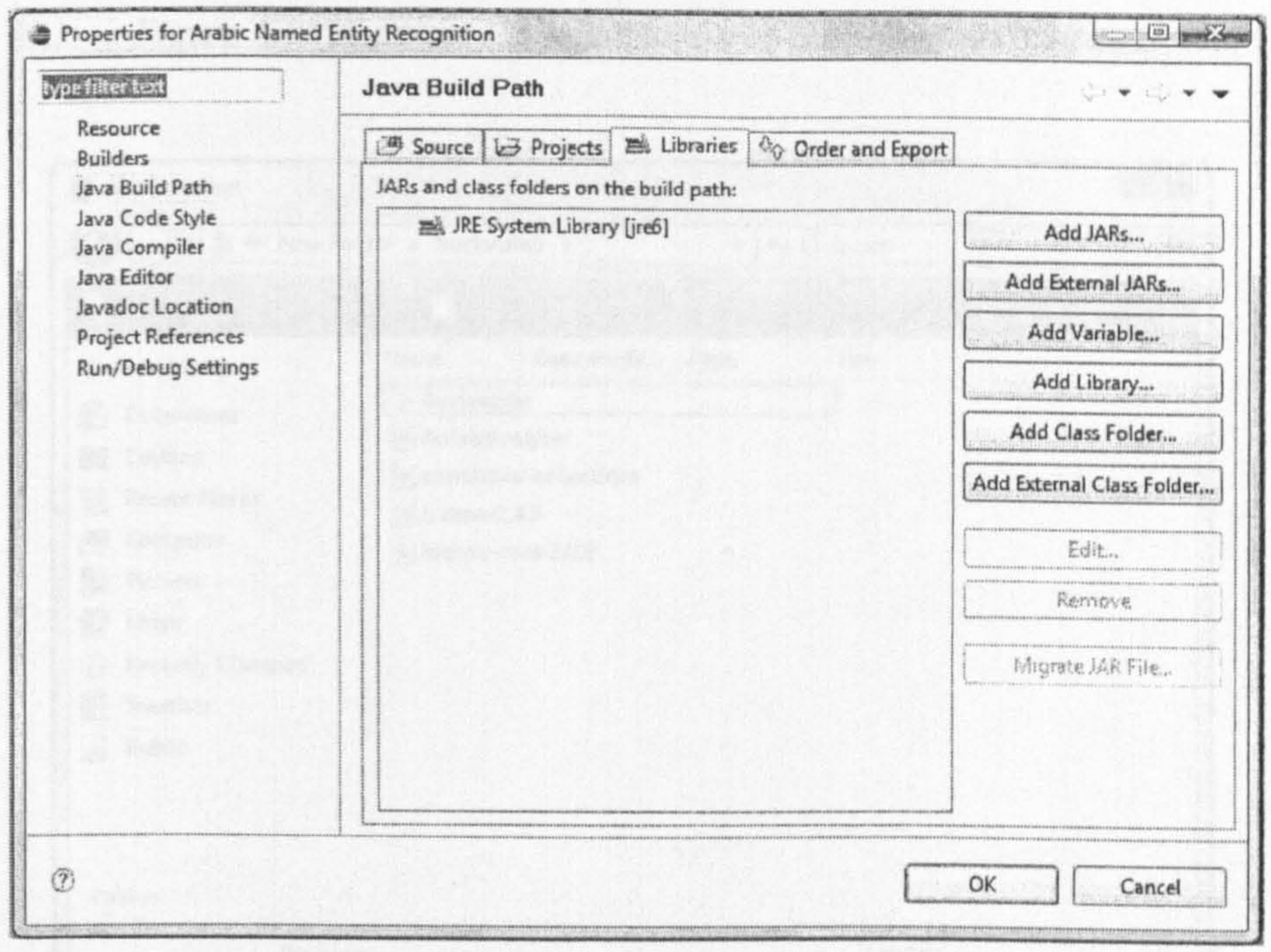

Figure 06.8: illustrate the integration of the systems with eclipse via JAVA library

We composed GATE, BAMA, and our JAVA classes and saved them in a particular place, therefore we located and imported them to our project "Arabic Named Entity Recognition" as shown in Figure 6.9, 6.10 and 6.11 respectively.

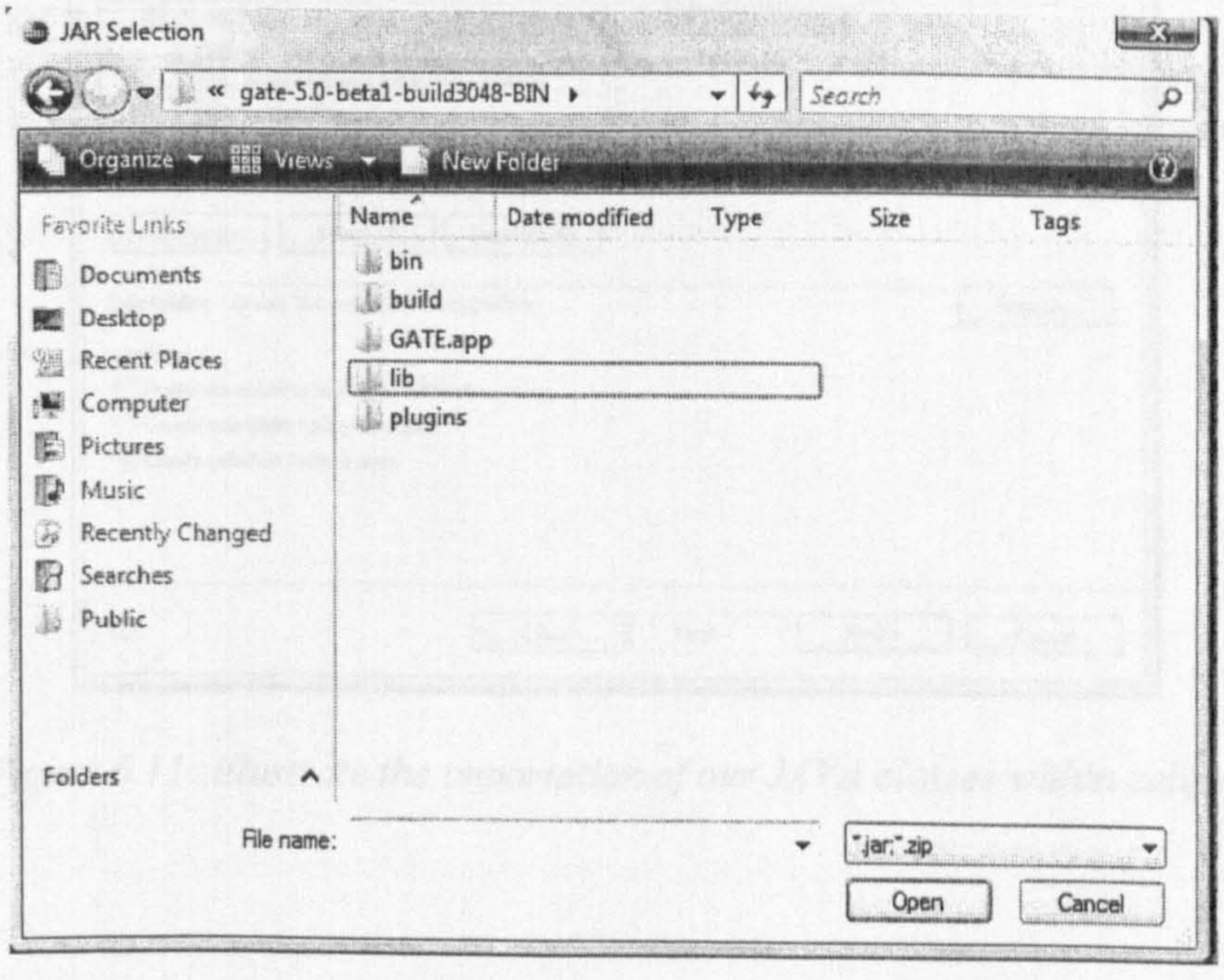

Figure 6.9: illustrate the importation of GATE within eclipse

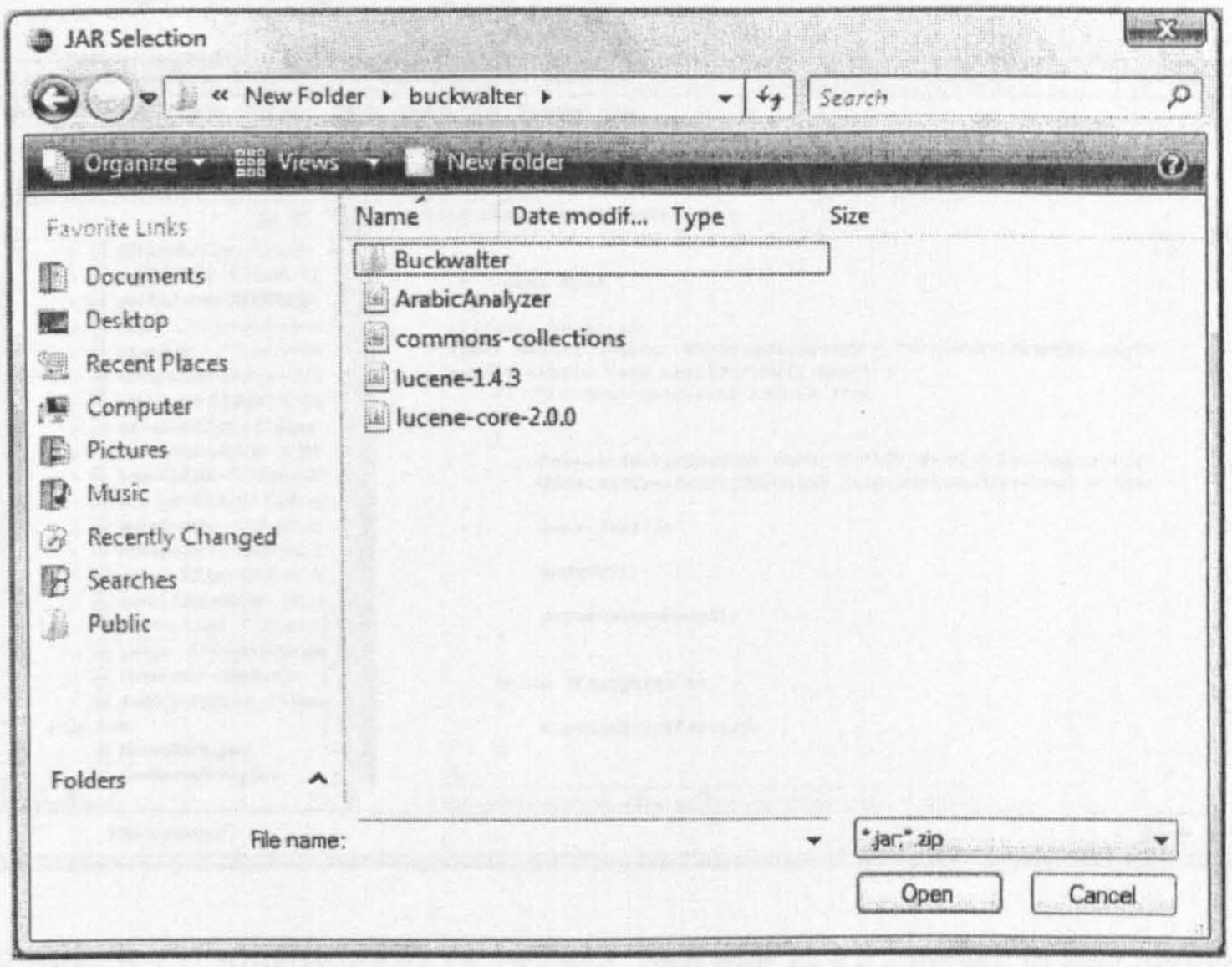

Figure 6.10: illustrate the importation of Buckwlter (BAMA) within eclipse

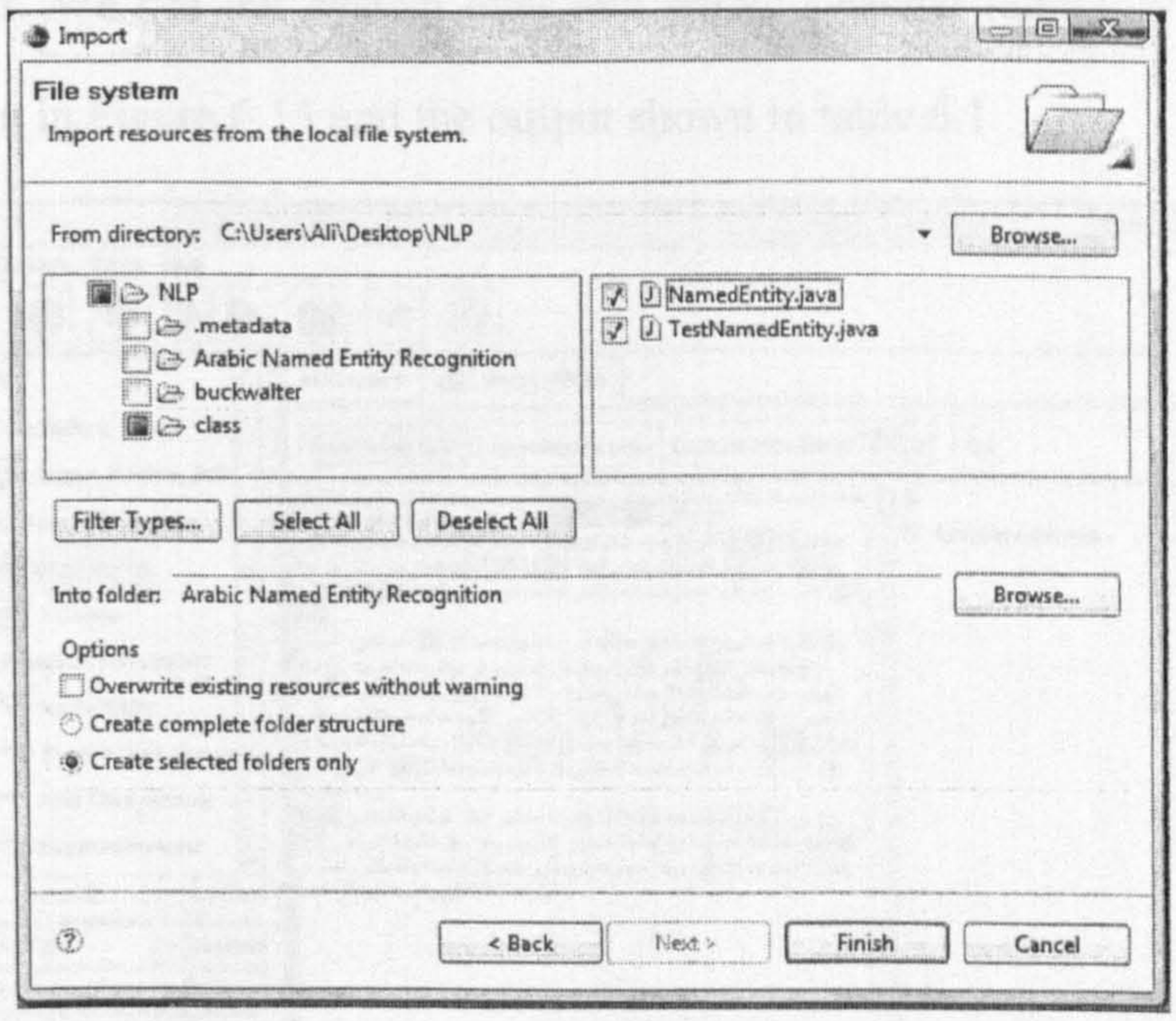

Figure 6.11: illustrate the importation of our JAVA classes within eclipse

The final step appears in figure 6.12 where GATE, BAMA, and our JAVA classes integrated and emerged through our project (Arabic Named Entity Recognition).

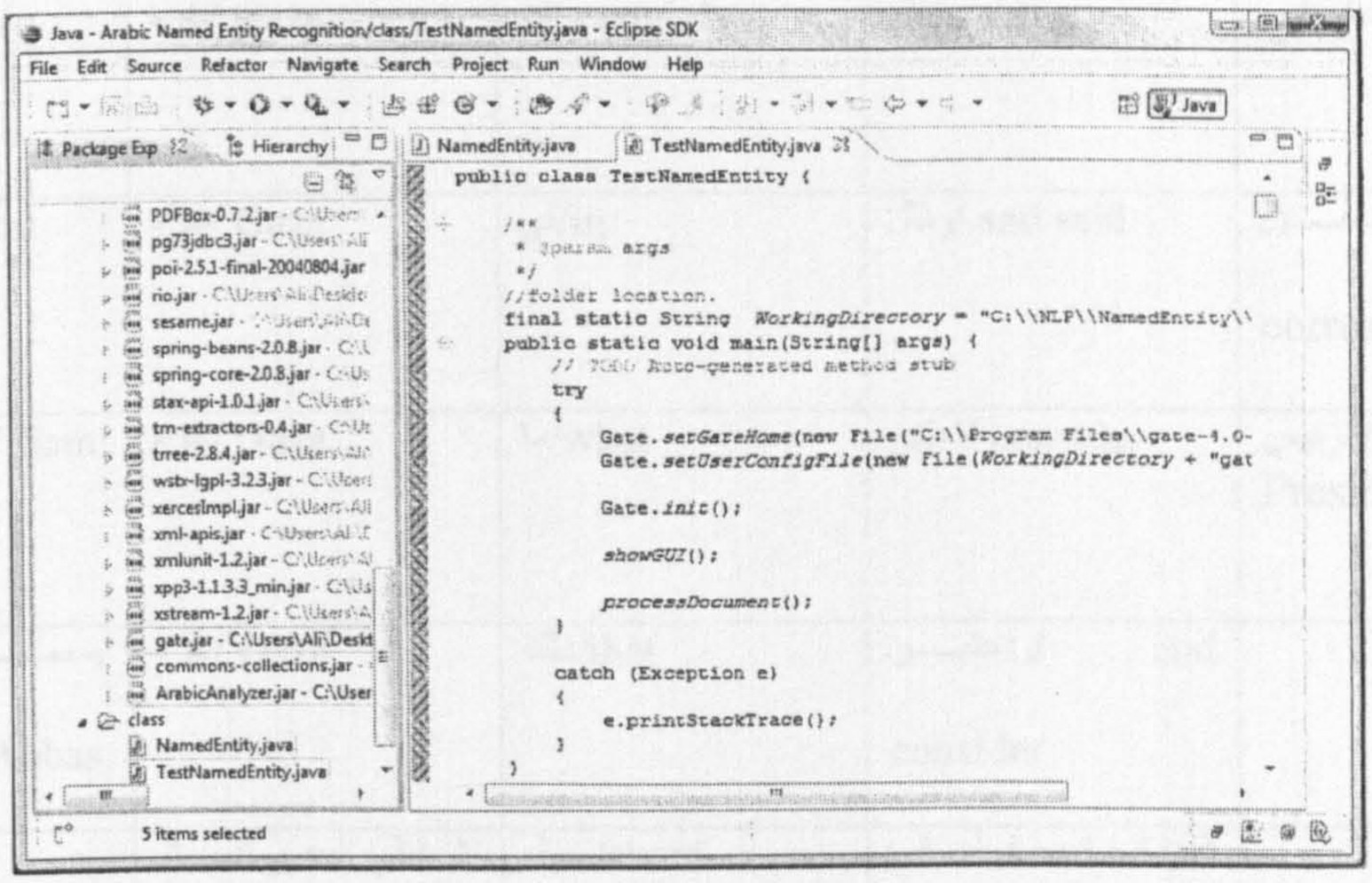

Figure 6.12: illustrate GATE, BAMA, and our JAVA classes in our project

At this stage our project is constructed and becomes ready to operate all the orders given by the developer. We run our system over one article selected randomly from Aljazeera website as shown in Figure 6.13 and the output shown in table 6.1.

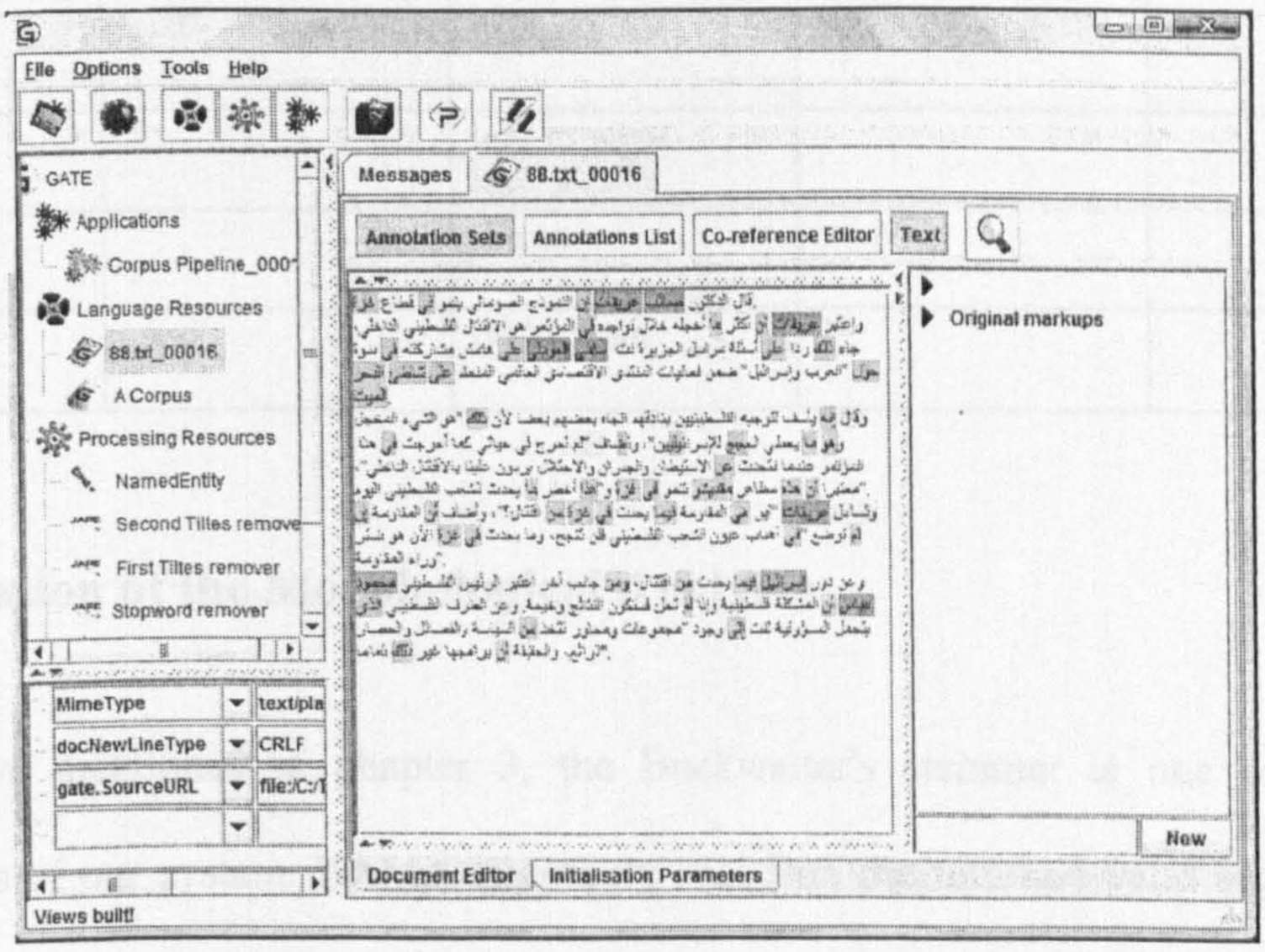

Figure 6.13: the experiment over one article.

Table 06.1: the output of the experiment given in figure 6.13

Person Name	***Location***	***Stop word***	***IVL***	***IWL***
صائب عريقات/ Saeb Erekat	غزة/ Gaza	إن/ certainly	قال/ said	الدكتور/ the doctor
عريقات/ Erekat	غزة/ Gaza	في/in	وقال/ and said	مراســـــــــل/ correspondent
سامي العودلي/ Sami Alaudli	غزة/ Gaza	ما/what	اعتبر/ consider	الرئيس/the President
محمـــــود عبـــاس/ Mahmoud Abbas	غزة/ Gaza	ذلك/that	واعتبـــر/ and consider	
	شـاطئ البحـر الميـت/ dead sea beach	حول/about	وأضاف/ and added	
	مقديشـــــــــــو/ Mogadishu	إنه/ certainly that	وتساءل/ and asked	
	إسرائيل/ Israel	عن/about		
		هذه/this		
		هذا/ this		
		فيما/ in what		
		لم/non		
		الذي/whom		

6.4 Discussion of the Morphological Analyzer

As we have mentioned in chapter 3, the Buckwalter's stemmer is one of the main components of our system. BAMA is entirely based on the returned valid segmentations given by the stemmer. However in some cases the stemmer returns incorrect or incomplete

result as the example given in Figure 6.14 shows. As a result of this issue the tag given to this result (stem) by BAMA will be incorrect.

```
SOLUTION #1
Lemma   :          mAlikiy~
Vocalized as :   AlmAlikiy~
Morphology :
        prefix : NPref-Al
        stem : Nall
        suffix : Suff-0
Grammatical category :
        prefix : Al      DET
        stem : mAlikiy~ NOUN
Glossed as :
        prefix : the
        stem : Malikite

SOLUTION #2
Lemma   :          mAlikiy~
Vocalized as :   AlmAlikiy~
Morphology :
        prefix : NPref-Al
        stem : Nall
        suffix : Suff-0
Grammatical category :
        prefix : Al      DET
        stem : mAlikiy~ ADJ
Glossed as :
        prefix : the
        stem : Malikite

==================== Statistics ====================
Lines : 1
Arabic tokens : 1
Non-arabic tokens : 0
Words found : 1 (100%)
Words not found : 0 (0%)
====================================================

C:\NLP\buckwalter>
```

Figure 6.14: BAMA incorrect stemmer solutions of the word (المالكي, Almalikiy)

In this example there are two solutions for the word (المالكي, Almalikiy) which are:

First solution:

Prefix: Al
Stem: malikiy Noun
Suffix: null

Second solution

Prefix: Al
Stem: malikiy adjective
Suffix: null

In both solutions the suffix is null. However, the word (المالكي, Almalikiy) in the Arabic language could be a proper noun if the last letter in the word (ي, iy) is considered as a suffix, then in addition to the two previous solutions a third solution is added as follows:

Third Solution

Prefix: Al
Stem: malik proper noun
Suffix: iy

From our experimentations and observations we noticed that the vast majority of the words stemmed correctly, however there are some words that are stemmed incorrectly as the example given above (around thirty words) and therefore we manually corrected and added them to the list where our system exclude them from the analyses of the stemmer accordingly the accuracy of the system slightly improved.

6.5 Summary

We illustrated in this chapter the design of our system and the communications of the systems component and we shows how the different tools and components are put together to develop our information extraction system using the JAVA programming language, and gave a real example of our experiment by running our system over one article. We highlighted the fact that sometimes BAMA returns incorrect results, which affects the accuracy of the results returned by BAMA. We developed rules to correct these inaccuracies.

Chapter 7

System Evaluation

7.1: Evaluation of IE Systems

The MUCs were the first attempt to standardize the task of IE and establish benchmark corpora; the goal of the MUC program was to provide a platform on which various IE approaches can be compared. There were seven MUC evaluations carried out in a ten years span, these evaluations greatly promoted the research in information extraction. The Automatic Content Extraction (ACE) evaluation program is a successor to the MUCs. The objective of the ACE Program is to develop extraction technology to support automatic processing of source language data (in the form of natural text), and eventually lead to the extraction of content from text at higher level. However Linguistic Data Consortium LDC develops annotation guidelines, corpora and other linguistic resources to support the ACE Program. Some of these resources have been developed in cooperation with the Translingual Information Detection Extraction and Summarization (TIDES) Program, in support of TIDES Extraction evaluations. The performance evaluation in IE systems is obtained by comparing the system result with a test answer keys that are produced manually (hand-tagged) (Lehnert et al, 1999). In the IE task, precision is the percentage of named entities found by the learning system that are correct, recall is the percentage of named entities present in the corpus that are found by the system and a named entity is correct only if it is an exact match of the corresponding entity in the data file. Hence the Precision and Recall are formulated as follows:

P= # of correct entities detected / # of entities detected.

R= # of correct entities detected / # of entities manually labelled.

Where # of correct entities detected represents the total number of the correctly extracted named entities; # of entities manually labeled represents the total number of named entities in the answer key test corpus and # of entities detected represents the total number of extracted entities (correct and incorrect). Both recall and precision are constants in the interval [0, 1], and their optimum being at 1.0 (Eikvil, 1999). From the field of IR the recall and precision measures are combined in a single accuracy measure called the F-measure (Makhoul et al, 1999) that is calculated as:

$$F = 2 P R / P+R$$

7.2 Problem with Evaluating Arabic IE Systems

Benajiba, Rosso, and Benedí (2007) conducted research on the Arabic NLP tools and resources in general (corpora, gazetteers, POS taggers, etc). They concluded that in comparison with other languages Arabic misses lexical resources, especially free resources available for research purposes. With respect to the corpora not many are available for the NER task. For instance, in the Conference on Natural Language Learning CoNLL 2002 the available corpora were only for the Chinese, English, French, Japanese, Portuguese and Spanish languages (Sundheim, 1995). There are no free Arabic corpora oriented to the NER task available (Benajiba, Rosso, and Benedí, 2007). This is the reason why we have decided to build our own corpora for training and testing our system.

7.3 Evaluation Methodology

We have developed our own corpus of native Arabic articles from Aljazeera website, initially we have manually tagged a small corpus (training corpora) consisting of 100 news articles. The training corpus was taken from politics domain. We manually labelled the entire person name on the training corpora. However through the analysis of the training corpora we have developed a number of rules, for instance we assumed that each word comes next to the trigger words considered as a person name, as the examples given in Table 7.1 where the word marked with a bold font represents the trigger word in the phrase.

Table 7.1: Examples of person names appearing next to a trigger word

The phrase	***The translation***
الرئيس حسني مبارك	**President** Hosni Mubarak
القائد معمر القذافي	**Leader** Muammar al-Gaddafi
السفير سالم نوري	**Ambassador** Salem Nouri
المدير عمر حسين	**Director** Omar Hussein
المذيع فيصل القاسم	**Announcer** Faisal Al-Qassem
الصحفي أحمد منصور	**journalist** Ahmed Mansur
القاضي سلمان أحمد	**Judge** Salman Ahmed

Consequently, we selected another 100 articles (test corpora) from the same site and we run our program over this corpus, we obtained the results as shown in Table 7.2.

Table 07.2: The initial results obtained by our system

	Recall	***Precision***	***F- Measure***
The person name	55%	49%	51%

We selected another 100 articles as new training corpus then we analyzed this new training corpus to improve our rules and accuracy of our system; we noticed that a lot of words come immediately next to the trigger word are (stop word) as shown in Table 7.3. Where the word marked with a bold font represents the stop word in the phrase.

Table 7.3: Examples showing stop words next to a trigger word

The phrase	***The translation***
قال إن	Said **that**
أكد على	Emphasize **for**
خرج من	Exit **from**
دخل إلي	Entered **to**
صرح بذلك	Announced **with**

As a result of this issue we collected and built a list of stop words and saved them as a gazetteer. We selected another 100 articles as a test corpus and we run our system over this corpus and we obtained the results shown in Table 7.4.

Table 07.4: System performance after using stop words.

	Recall	***Precision***	***F Measure***
The person name	63%	60%	61%

Then, we used another 100 articles as a training corpus and manually marked the entire person names available in the training corpora and noticed that the person name sometimes is located after several words from the trigger word. Moreover, some of these words begin with the definite article (ال, the) as shown in Figure 7.5 where the words marked with a bold font represents words beginning with the definite article (ال, the). We selected another 100 articles as a test corpus and we applied the rules we presented in chapter 5, section 2.1.3.

Table 07.5: Example of words begining with (ال, the)

The phrase	*The translation*
رئيس الوزراء السابق الفرنسي	**Former French** Prime **Minister**
السفير الأمريكي الحالي	**The current American**. ambassador
وزير الدفاع العراقي السابق	**Former Iraq defence** minister
أمين الجامعة العربية	**Arab League** Secretary

Afterwards we run our program over the new test corpora and we obtained the results shown in Table 7.6.

Table 07.6: System performance after considering words beginning with (ال, the) isolated

	recall	***precision***	***F- Measure***
The person name	*69%*	*64%*	*66%*

We reiterated this procedure several times and each time the performance of the system improved. However the vast impact to the performance of our system occurred when we combined the rules developed in chapter 5 with the BAMA system where we scanned all the output of BAMA and then we selected the correct result and as a result the F- measure of our system increased from 66% to 89.67%, the final results obtained from our final experiment will be discussed in next chapter.

7.4 Comparison of our System with other System

As we mention early, Arabic lacks linguistic resources, especially free resources for research purposes, therefore we constructed our own corpora for our system. However for comparison purposes we got in touch with several researchers working in the Arabic NER field (Benajiba et al, 2007), (Shamsi and Guessoum, 2006), (Shaalan, 2007) and (Mesfar, 2007) to compare our system with their systems but unfortunately we did not get any

response from them except from (Mesfar, 2007). Mesfar (2007) developed a system for the recognition of proper names, dates, and numerics in standard Arabic text through a combination of a morphological analysis and a rule-based NER system using NooJ syntactic grammars, the architecture of the system shown in Figure 7.1. NooJ is a linguistic developmental environment used to formalize various types of textual phenomena. NooJ includes tools to construct, test, debug, maintain and accumulate large sets linguistic resources, and can apply them to large texts. Moreover dictionaries and grammars are applied to texts in order to locate morphological, lexicological and syntactic patterns, solve ambiguities, and tag simple and compound words.

The author has constructed his own corpus based on the "Le Monde Diplomatique" newspaper, Arabic version. Being unable to have a copy of his corpora, we have instead send our corpora to the author who has kindly used his system to extract the named entities. We have used these results to compare his system against ours. The results of this comparison are shown in Table 7.7.

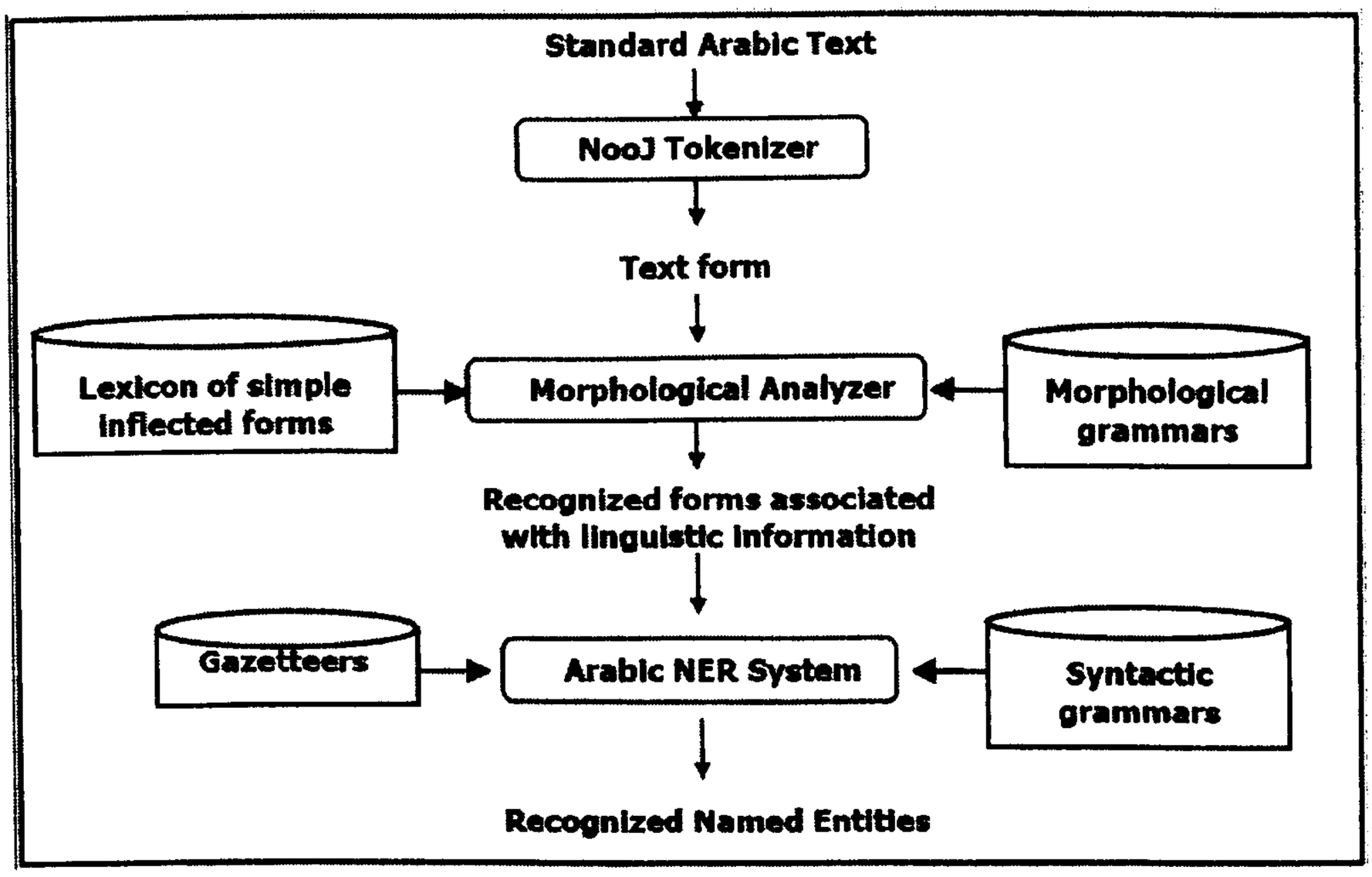

Figure 07.1: Mesfar's system architecture (Mesfar, 2007).

Table 07.7: Comparison of Mesfar's results and our results (in bold and italic font)

	Precision	***Recall***	***F- measure***
Person name	***93.2%***	***86.4%***	***89.67%***
	95.6	90.7%	93.08
Organization	***87.9%***	***85.2%***	***86.52%***
	89.5%	78.8%	83.80%
Location	***93.6%***	***78%***	***85.87%***
	94.2%	75.7%	83.94%
Time	***96.7%***	***95.6%***	***96.14%***
	–	–	–
Date	***95.4%***	***92.3%***	***93.82%***
	–	–	–
Money	***93.9%***	***95.3%***	***94.59%***
	97.6%	96.1%	97.34%

It is worth noting that we did not have any control over the results of Mesfer's system but relied only on the results we were given. The results are very close except for person name, where his results are slightly better than our results and the reason behind this is his use of a huge names gazetteer with over 12400 persons name entries unlike our system which entirely rely on the developed rules. It is worth noting that we did not get any results for the Date and Time Entities and therefore could not perform any comparison.

7.5 Summary

In this chapter we described how information extraction systems are evaluated and defined the methodology we used to evaluate our system. Given that there are no free corpora for the Arabic language, we constructed our own corpora based on news articles extracted from Aljazeera website. The overall performance of our system improved when including

BAMA. Moreover we evaluated our system with other system using the same corpora and we gave overview of the comparison of the results. Finally, we gave overview of several systems with regarding to the methods, the corpora which have been used, and the results which have been obtained. We summarized our reviewing in Table 7.8.

Table 07.8: Summary of the performances of Arabic NER systems

Reference	***Method***	***Corpus used***	***Precision***	***Recall***	***F-Measure***
Saleem (2004)	Rule Based	500 articles from the Al-Raya newspaper (2003), published in Qatar			78.4
Samy et al. (2005)	Parallel corpus and reusing previously developed (PoS) for other languages (Spanish)	(1200 sentence pairs) An Arabic-Spanish parallel corpus	84%	97.5%	90%
Zitouni et al.(2005)	Statistical Maximum Entropy	ACE 2003	75.3%	70.2%	72.7%
Shamsi and Guessoum (2006)	Statistical Hidden Markov Model (HMM).	Own corpus from various Arabic article	97%	–	–
Shaalan and Raza, (2007)	Rule Based	ACE and Treebank corpus	91.2%	90.3%	90.7%
Benajiba, Rosso, & BenedÃ (2007)	Statistical maximum entropy	Own corpus 316 articles from several newspaper	63.21%	47.58%	54.11%
Benajiba and Rosso, (2008)	Statistical Support Victor Machine	ACE data and a manually created data set UPC-corpus.	82.71%	80.4%	79.21%

Chapter 8

Conclusion and Future work

In the conclusion of this thesis, we summarize the scope of this work, the approach and the results achieved while delivering the outcome of this thesis. Then, we highlight the main developments for future work.

8.1 Conclusion

The work described in this thesis concerns IE and more specifically, named entity extraction in Arabic. However, with the huge amount of published data in Arabic we recognize that developing a system to extract important data from documents becomes essential. The Arabic language is a language of significant interest to the NLP community mainly due to its political and economic significance, but also due to its interesting characteristics. There are two basic approaches to IE system development: the Rule based (Knowledge Engineering) Approach and the Automatic Training Approach (statistical). After we investigated the characteristics of the Arabic language and the approaches used in the field of the NE recognition we concluded that because of the nature of the Arabic language, the rule based system is more appropriate than the statistical system to identify Arabic NE. Our system architecture is based on the GATE system as there is no other software in the NLP field with the robustness and characteristic of GATE that is freely available. In addition, BAMA is built over GATE to achieve our goals. BAMA is widespread and heavily used in the literature. BAMA depends entirely on the embedded stemmer; however the output of the stemmer given a tag by BAMA such as noun, verb, pronoun, proper noun, etc. in some cases the stemmer return incorrect results, and this issue affects the accuracy of the results given by BAMA. Therefore we manually corrected

these incorrect results and we obtained more accurate results. Moreover, in order to tag named entity in Arabic texts, lists of trigger words have been identified to find the place where we can locate proper names in the text. By using keywords we mark name phrases that might include a certain name then we process these phrases to extract these names. Therefore we developed and implemented our rules in order to recognise each position of the NE. All the programs we have used in our application are implemented using the JAVA programming language; consequently we selected eclipse as the integrated development environment to develop our application. Similarly, it is not possible to find free Arabic corpora oriented to the NER task. Thus, we have decided to build our own corpora for training and testing to carry out this work. We collected corpora from Aljazeera website and we run our system over these corpora. We reiterated this process until we analyzed and classified one thousand articles. We have also compared our system with other Arabic named entity recognition systems developed using both rule based and statistical approaches. Our system achieves results better or similar results than the other systems.

8.2 Future Work

The domain scope that has been used to conduct our experiment was about political events. This required us to collect our trigger word and implement our rules in a way that reflects the characteristics of the texts in this domain. However, we intend to extend our work to include a number of other domains such as economic, sport, religion, etc. We believe that changes will be minimal and all what is need is the collection of new trigger words and adding new rules. We also need to collect and develop larger and independent electronic corpora. This corpus has to be divided into two sections. One section for building and testing the NER systems, and the other one for evaluation of these systems.

Appendix A: Qalam Transliteration Scheme

Arabic Letter	Description	Transliteration	Arabic Letter	Description	Transliteration
ء	hamza	'	ط	Taa'	T
ا	'alef	aa	ظ	Zaa'	Z
ب	baa'	b	ع	`ayn	`
ت	taa'	t	غ	ghayn	gh
ث	thaa'	th	ف	faa'	f
خ	jym	j	ق	qaaf	q
ح	Haa'	H	ل	laam	l
خ	khaa'	kh	ن	nuwn	n
د	daal	d	م	meem	m
			و	waaw	w
ذ	dhaal	dh	ك	kaaf	k
ر	raa'	r	ه	haa'	h
ز	zayn	z	ي	yaa'	y
س	syn	s	ة	taa'	t
ش	shyn	sh	ه	haa'	h
ص	Saad	S	ى	'alef	ae
ض	Daad	D			
Diacritics (tashkyl)					
فَ	fatHah	a	آ	maddah	~aa
فِ	kasrah	i	فّ	shaddah	Consonant
فُ	Dammah	u	اً	tanwyn	N
			فْ	sukuwn	-

Appendix B: My Publications

Elsebai, A., (2008). Arabic Proper Names Recognition Using Heuristics, Proceedings of the 9th Annual Postgraduate Symposium 0n Convergence of Telecommunications, Networking and Broadcasting, 86- 88, Liverpool, UK, 23-24 June 2008.ISBN: 978-1-902560-19-9.

Elsebai, A., & Meziane, F., (2008). Recognize Person Names from Arabic Text based on Rule Based, Proceeding of the 4th Informatics Research Institute Postgraduate Conference, pp 151- 153, Salford, UK, ISBN: 978-1-905732-57-9.

Elsebai, A., & Meziane, F., (2008). Extracting Persons Names from Arabic Newspaper, Proceedings of the 5th international conference on Innovations in Information Technology, Dubai, United Arab Emirates, 16-18 December 2008.

Elsebai, A., Meziane, F., & Belkredim, F., Z., (2009), A Rule Based Persons Names Arabic Extraction System, Proceedings of the 11th International Business Information Management Association Conference (IBIMA 2009), Special Track on Arabic Information Processing, pp. 1205-1211, Cairo, Egypt 4-6 January 2009. ISBN 978-0-9821489-0-7.

Belkredim, F.Z. & Elsebai, A., (2009). An Ontology Based Formalism for the Arabic Language Using Verbs and their Derivatives, Proceedings of the 11th International Business Information Management Association Conference (IBIMA 2009), Special Track on Arabic Information Processing, pp. 1196-1204, Cairo, Egypt 4-6 January 2009. ISBN 978-0-9821489-0-7.

Elsebai, A., & Meziane, F., (2009). Named Entity Recognition in Arabic: A Review of Some Current Systems, Proceedings of the 12th International Business Information Management Association Conference (IBIMA 2009), Kuala Lumpur, Malaysia 29-30 June 2009. ISBN: 978-0-9821489-1-4

Bibliography

Abraham, A, (2005). Rule Based Expert Systems, Handbook for Measurement Systems Design, Peter Sydenham and Richard Thorn (Eds.), John Wiley and Sons Ltd., London, ISBN 0-470-02143-8, pp. 909-919, 2005.

Abuleil, Saleem, 2004. "Extracting Names From Arabic Text for Question-Answering Systems". RIAO'04, Proceeding of the 7th International Conference on Coupling Approaches, Coupling Media, and Coupling Languages For Information Retrieval. University of Avignon (Vaucluse), France April 26th-28th, 2004. pp. 638-647.

Abuleil, S. and Evens, M., (2002). Extracting an Arabic Lexicon from Arabic Newspaper Text. Computers and the Humanities, 36(2), pp. 191-221.

Abu-Salem, Hani, Al-Omari, Mahmoud, Evens, Martha.(1999). Stemming Methodologies over Individual Query Words for an Arabic Information Retrieval System. JASIS 50(6): 524- 529, 1999.

Al-Daimi, K., and Abdel-Amir, M. (1994). The Syntactic Analysis of Arabic by Machine. Computers and Humanities, Vol. 28, No. 1, pp. 29-37.

Aljazeera TV, http://www.aljazeera.net/, 2008

Aljohar, B., (1999). A Portable Natural Language Interface from Arabic to SQL, PhD thesis, University of Sheffield, UK, 1999.

Allen, J. (1987). Natural language understanding. The Benjamin/Cummings Publishing Company, Inc.

Al-Fedaghi Sabah S. and Fawaz Al-Anzi (1989). A new algorithm to generate Arabic root-pattern forms. Proceedings of the 11th National Computer Conference, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia., pp04-07

Al-Kharashi, I. and Evens, M. W.(1994). Comparing words, stems, and roots as index terms in an Arabic information retrieval system. JASIS, 45 (8), pp. 548-560, 1994.

Al-Shalabi, R. and Evens, M., (1998). .A Computational Morphology System for Arabic. Workshop on Semitic Language Processing. COLING-ACL.98, University of Montreal, Montreal, PQ, Canada, Aug 16 1998. pp. 66-72.

Andrew, R., 2003. Machine Learning in Natural language Processing. http://www.andy-roberts.net/misc/latex/sessions/bibtex/bib_example_nat.pdf, accessed 23/06/2008.

Appelt, D. E. & Israel, D. (1999). Introduction to Information Extraction Technology. International Joint Conference on Artificial Intelligence Tutorial, Sweden.

Asahara, M. & Matsumoto, Y (2003). Japanese Named Entity Extraction with Redundant Morphological Analysis. In Proceedings of Human Language Technology Conference(HLT-NAACL).

Attia, M. (2006). An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks. The Challenge of Arabic for NLP/MT Conference. The British Computer Society, London, UK.

Babych, B. & Hartley, A. (2003). Improving Machine Translation Quality with Automatic Named Entity Recognition: In Proceedings of the 7th International EAMT workshop on MT and other language technology tools. Improving MT throught other language technology tools. Recourses and tools for building MT. Budapest, Hungary. p. 1–8.

Benajiba, Y., Rosso, P., Bened´ı, J. (2007). ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy. In: Proc. 8th Int. Conf. on Comput. Linguistics and Intelligent Text Processing, Springer-Verlag, LNCS (4394), pp. 143-153.

Benajiba, Y., and Rosso. P. (2008). Arabic Natural Language Processing Session (ANLP-ACIT'2008) at the International Arab Conference on Information Technology (ACIT'2008), Hammamet, December 16-18, 2008

Benajiba, Y., Diab M., Rosso P. (2008). Arabic Named Entity Recognition using Optimized Feature Sets. In: Proc. Int. Conf. on Empirical Methods in Natural Language Processing, Waikiki, Honolulu, U.S.A.

Bennett, S. Aone, C & Lovell, C (1997). Learning to tag multilingual texts through observation. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, Rhode Island (1997), pp:109-116.

Berri, J., Zidoum, H., & Atif. Y., (2001). Web-based Arabic Morphological analyser. In conferences on Computational Linguistics and Intelligent Text Processing, Germany (2001), pp: 216-225.

Bikel, D.M., Miller, S., Schwartz, R. & Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In Proceedings of Applied Natural Language Processing, 194–201.

Bikel, D. M. Schwartz, R. & Weischedel, R. M. (1999). An Algorithm that Learns What's in a Name, Machine Learning, Vol. 34, No. 1-3, (1999), pp: 211-231

Black, W. J.; Rinaldi, F, Mowart, D., (1998). FACILE: Description of the NE System Used for MUC-7. In Proceedings of the MUC-7, 1998.

Borthwick, A. Sterling, J. Agichtein, E. & Grishman, R. (1998). Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In Proceedings of the Sixth Workshop on Very Large Corpora, Canada (1998), pp: 152-160.

Brill, E., (1994). Some Advances in Transformation Based Part of Speech Tagging. In "Proceedings of the Twelfth International Conference on Artificial Intelligence" (AAAI-94), Seattle, WA.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics, 21, 543–565.

Brill, E., (1992). A simple rule-based part of speech tagger. In Proceedings of the DARPA Speech and Natural Language Workshop. Harriman, NY, 1992.

Buckwalter, T. (2004). Buckwalter Arabic Morphological Analyzer, version 2.0. LDC catalog No LDC2004L02, Linguistic Data consortium, www.ldc.upenn.edu/Catalog.

Buchholz, S., (2002). Memory-Based Grammatical RelationFinding. Ph.D. thesis, University of Tilburg.

Califf, M., E., and Mooney, R., (1997). Applying ILP-based Techniques to Natural Language Information Extraction: An Experiment in Relational Learning. In Working Notes of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming. Nagoya, Japan, August, 1997.

Chen, A., & Gey, F. (2002). Building an Arabic stemmer for information retrieval. In TREC 2002. Gaithersburg: NIST, pp 631-639, 2002.

Chinchor, N., (2001).Overview of MUC-7/MET-2. Retrieved 23 June 2008 from, http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html

Chinchor, N., & Sundheim B. (1993). MUC-5 Evaluation Metrics. In Proceedings of the Fifth Message Understanding Conference (MUC-5), pages 69-78. Morgan Kaufmann.

Ciravegna, F., Campia, P., & Colognese(1992), A. Knowledge Extraction from Texts by SINTESI. In: Proceeding of the Fifteenth International Conference on Computational Linguistics (COLING-92). 1992, 1244-1248.

Church, K., (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In "Proceedings of the Second Conference on Applied Natural Language Processing" (ACL) Austin, Texes,pp. 136-143

Ciravegna, F., Lavelli, A., Gilardoni, L. Mazza,. S. Black, W., J., Ferraro, M., Mana,. N., Matiasek, J., Rinaldi., F., (2000). Flexible Text Classification for Financial Applications: The FACILE System. In Proceedings of Prestigious Applications sub-conference (PAIS2000) sub-conference of the 14th European Conference On Artificial Intelligence (ECAI2000), Berlin, Germany, August, 2000.

Cowie, J., & Lehnert, W. (1996). Information extraction. Communications of the ACM, 39(1), pp. 80-91.

Cunningham, H., Maynard, D., & Tablan, V., (2000). JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November.

Cunningham, H., Gaizauskas, R., & Wilks Y., (1995). A General Architecture for Text Engineering (GATE) – a new approach to Language Engineering R&D. Technical Report CS–95–21, Department of Computer Science, University of Sheffield, 1995.

Cunningham, H. (2000). Software Architecture for Language Engineering. Computer Scinece. Sheffied, Sheffield. PhD.

Cunningham., H., Maynard., D., Bontcheva., K., Tablan., V., Ursu., C., Dimitrov., M., Dowman., M., Aswani., N., Roberts., I., Shafirin., Y., Li., A., Funk., A., (2010). Developing Language Processing Components with GATE Version 5.2), http://gate.ac.uk/sale/tao/split.html [accessed 17 February 2010].

Cutting, D., Kupiec, J., Pederson, J., & Sibun P. (1992). A Practical Part-of-Speech Tagger. In "Proceedings of the Third Conference on Applied Natural Language Processing", Trento, Italy.

DARPA (Ed), (1998). Proceeding of the Seventh Message Understanding Evaluation and Conference (MUC-98), Fairfax, VA. Morgan Kaufman.

Darwish, K., (2002). Building a shallow Arabic morphological analyser in one day. In Proceedings of the Workshop on Computational Approaches to Semitic Languages, USA (2002), pp: 22-29.

Diab M., Hacioglu K. & Jurafsky D. (2004). Automatic Tagging of Arabic Text : From Raw Text to Base Phrase Chunks. In proc. of HLTNAACL'04 (Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics): 149–152.

Ditters E. (2001). A Formal Grammar for the Description of Sentence Structure in Modern Standard Arabic, In the proceeding of Arabic NLP Workshop at ACL/EACL.

Dunphy., G., and Metwally., A., (2006). Pro BizTalk 2006 Publisher: Press Print ISBN: 978-1-59059-699-9, page 286.

Eikvil, L. (1999). Information extraction from world wide web - a survey. Technical Report No 945, ISBN 82-539-0429-0, Norweigan Computing Center, Norway.

EI-Sadany, T. A., & Hashish, M. A. (1989). An Arabic Morphological System. IBM Systems Journal. Vol.28, No.4, 600-612.

Ernest F. H. (2003). Jess in Action Java Rule-based Systems, Manning Publication, Sandia National Labs, ISBN: 1930110898 (2003).

Freeman A.T. (2001). Brill's POS tagger and a Morphology parser for Arabic. In proc. Of ACL'2001 (the 39th Annual Meeting of Association for Computational Linguistics & 10th Conference of the European Chapter, Workshop on Arabic Language Processing): 7.

Fukuda, K., Tamura, A., Tsunoda, T., & Takagi, T. (1998) Toward information extraction: identifying protein names from biological papers. PSB'98. pp 707-18.

Garside, R., Leech G., & Sampson., G. (1987) The Computational Analysis of English: a corpus-based approach. Longman Group UK Limited.

Gaizauskas, R. & Yorick W. (1998). Information Extraction: Beyond Document Retrieval. Journal of Documentation. 54, no. 1 (January 1998): pp. 70-105.

Greene, B., B., & Rubin, G.M. (1971) Automatic Grammatical Tagging of English. Department of Linguistics, Brown University, Providence, R.I.

Greenwood, M. & Gaizauskas R. (2007). Using a Named Entity Tagger to Generalise Surface Matching Text Patterns for Question Answering. In Proceedings of the Workshop on Natural Language Processing for Question Answering (EACL03).

Grishman, R. & Sundheim B. (1996). Message Understanding Conference-6: A Brief History. In Proc. International Conference on Computational Linguistics.

Grishman, R. (1997). Information Extraction: Techniques and Challenges. Information Extraction (International Summer School SCIE-97), Springer-Verlag.

Hajic J, S. O., Buckwalter T, Jin H (2005). Feature-Based Tagger of Approximations of Functional Arabic Morphology. The Fourth Workshop on Treebanks and Linguistic Theories. Universitat de Barcelona

Harmain, H.M. & Badr Aljohar (2001). A Framework for an Arabic Information Extraction System, The Proceedings of the Second Arab Conference on IT, 13-15-11/2001, Jordan.

Harris, Z. S. (1957). Linguistic Transformations for Information Retrieval. In Proceedings of International Conference on Scientific Information.

Hegazi, N., & EISharkawi, A. A. (1986). Natural Arabic Language Processing, Proceedings of the 9th National Computer Conference and Exhibition, Riyadh, Saudi Arabia, 1-17.

Hepple, M., (2000). Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), Hong Kong, October 2000.

Hiemstra, D. (2001). Using Language Models for Information Retrieval. PhD thesis, University of Twente. ISBN 90-75296-05-3.

Hobbs, J., R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., & Tyson. M., FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Finite State Devices for Natural Language Processing, MIT Press, USA, (1996) pp: 383—406.

Huang, F. (2005). Multilingual Named Entity Extraction and Translation from Text and Speech. Ph.D. Thesis. Carnegie Mellon University.

Humphreys K, Demetriou G, Gaizauskas R (2000). Two Applications of Information Extraction to Biological Science Journal Articles: Enzyme Interactions and Protein Structures. PSB'2000, 5:502-513.

Jaccarini A. (2001). A modifiable structural editor of grammars for Arabic processing, In the proceeding of Arabic NLP Workshop at ACL/EACL.

Jackson, P. & Moulinier, I. (2002). Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization, volume 5 of Natural Language Processing. John Benjamins Publishing Co., Amsterdam, 1st edition.

Janet C.E. (2002). The Phonology and Morphology of Arabic. The Phonlogy of the World's Languages. Oxford University Press, 2002.

Jurafsky., D., & Martin., J. (2000). Speech and Language Processing. Prentice Hall, USA (2000), pp: [191-219,235-283].

Kadri Y. & Benyamina A. (1992). 'A syntax semantic analyzer for Arabic language'.Engineer thesis, University of Oran 1992

Kaiser, K., & Miksch, S (2005). Information Extraction: A Survey. Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Technical Report, Asgaard-TR-2005-6, May 2005.

Kamir D, Soreq N, Neeman Y (2002): A Comprehensive NLP System for Modern Standard Arabic and Modern Hebrew, Proceedings of the workshop on Computational Approaches to Semitic Languages in the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02). Philadelphia, PA, USA.

Kashani, M,. Popowich, F, & Sadat. F. (2005). Automatic Transliteration of Proper Nouns from Arabic to English, School of Computing Science, Simon Fraser University, National Research Council of Canada.

Khoja S. (2001). APT: Arabic Part-of-speech Tagger. In proc. of NAACL'2001 (the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics): 20-26.

Kokkinakis, D. (1998). AVENTINUS, GATE and Swedish Lingware. In Proc. of Nordic Computational Linguistics Conference.

Larkey, L. & AbdulJaleel, M. (2003). What's in a Name? Proper Names in Arabic Cross Language Information Retrieval. CIIR Technical Report, IR-278. Dept. of Computer Science, USA.

Larkey, L., Ballesteros, L., & Connell. M., (2002). Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In SIGIR 2002, pages 275–282.

Larkey, L., Ballesteros, L., & Connell, M. (2007). Light Stemming for Arabic Information Retrieval: Knowledge-based and Empirical Methods, A.Soudi, A. van den Bosch, and Neumann, G., Editors. Kluwer/Springer's series on Text, Speech, and Language Technology. 2007

LDC, Linguistic Data Consortium. (2002).. Buckwalter Morphological Analyzer Version 1.0, LDC2002L49, 2002.

Lovins, J. B.(1968). Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11, pp. 22- 31, 1968.

Makhoul, J., Kubala, F., Schwartz. R & Weischedel, R (1999). Performance measures for information extraction. In Proc. of the DARPA Broadcast News Workshop, USA (1999).

Maloney, J., & Niv, M. (1998). TAGARAB: A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. In Workshop on Computational Approaches to Semitic Languages, Malta.

Marques., N. & Lopes., L. (1996) Using Neural Nets for Portuguese Part-of-Speech Tagging. In "Proceedings of the Fifth International Conference on the Cognitive Science of Natural Language Processing", Dublin City University.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., & Wilks, Y. (2001). Named Entity Recognition from Diverse Text Types. In Recent Advances in Natural Language Processing Conference, pages 257–274, Tzigov Chark, Bulgaria.

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, pp 305-316, Paris, France.

Meziane, F. (1994). From English to Formal Specifications, PhD Thesis. Department of Mathematics and Computer Science, University of Salford.

Mikheev, A. & Grover, C (1998). LTG: Description of the NE recognition system as used for MUC-7. In Proceedings of the Seventh Message Understanding Conference.

Moisiadis, F. Genrich, R. Stair, R. Reynolds, G. (2008). Principles of information systems, ISBN 9780170132831, pp 481.

NGAI, G. & FLORIAN, R. (2001). Transformation-based learning in the fast lane. In Proceedings of NAACL'01, 40–47, Pittsburgh, PA.

Piltrosanti, E., & Graziadio, B. (1997). Artificial Intelligence and Legal Text Management: Tools and Techniques for Intelligent Document Processing and Retrieval. In: Natural Language Processing: Extracting Information for Business Needs. Unicom Seminars Ltd., London, March 1997, 277-291.

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3):130-137, 1980.

Popovic, M. & Willett, P. (1992). The effectiveness of stemming for natural-language access to Slovene textual data. JASIS, 43 (5), pp. 384-390, 1992.

Pouliquen, B., Steinberger, R., Ignat, C., Temnikova, I., Widiger, A., Zaghouani, W., & Zizka. J. (2005). Multilingual person name recognition and transliteration. Journal CORELA-Cognition, Représentation, Langage, Vol. 2, ISSN 1638-5748.

Proux D, Rechenmann F, Julliard L, Pillet V, Jacq B (1998). Detecting Gene Symbols and Names in Biological Texts. A First Step toward Pertinent Information Extraction. Genome Informatics. 9:72-80.

Rau, L. (1991). Extracting Company Names from Text. Proceedings of the Seventh Conference on Artificial Intelligence Applications, Feb. 24-28, Miami Beach, Florida, pp.29-32.

Riloff, E. (1996). Automatically Generating Extraction Patterns from Untagged Text. Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96) , 1996, pp. 1044-1049.

Rindflesch, Thomas C.; Lorraine Tanabe; John W. Weinstein; and Lawrence Hunter. (2000). EDGAR: Extraction of drugs, genes and relations from the biomedical literature. Pacific Symposium on Biocomputing.

Ritchey, T., (1998). Morphological Analysis - A general method for non-quantified modeling. Adapted from a paper presented at the 16th Euro Conference on Operational Analysis, Brussels, July 1998.

Roeck, A. and Al-Fares, W., (2000). .A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong, Oct 1-8, 2000. pp.199-206.

Rogati, M., McCarley, S., & Yang, Y. (2003). Unsupervised learning of arabic stemming using a parallel corpus. In Proceedings of ACL-2003, Sapporo, Japan 2003 (pp. 391–398).

Sabri, E., William B., Piek V., David F., Adam P., & Christiane F., (2006). Arabic WordNet and the Challenges of Arabic. The Challenge of Arabic for NLP/MT. International conference at the British Computer Society, London, 23 October 2006; pp.15-24. [PDF, 295KB]

Sager, N., Friedman, C., & Lyman, M. (1987). Medical Language Processing: Computer Management of Narrative Data. Addision Wesley.

Sager, N. (1981) Natural Language Information Processing. Reading, Massachusetts: Addison Wesley, 1981.

Samy, D., Moreno, A., & Guirao, J.M. (2005). A Proposal for an Arabic Named Entity Tagger Leveraging a Parallel Corpus (Spanish-Arabic). In Proceedings of International Conference on Recent Advances on Natural Language Processing RANLP 2005, Borovets, Bulgaria, pp. 459-465.

Sekine, s. Grishman, R. & Shinnou, H. (1998). A decision tree method for finding and classifying names in Japanese texts. In Proceedings the Sixth Workshop on Very Large Corpora, 1998.

Selkov, E., Basmanova, S., Gaasterland, T., Goryanin, I., Gretchkni, Y., Meltsev, N., Nenashev, V., Overbeek, R., Panyushkina, E., Pronevitch, L., & Yunis., I. (1996). The metabolic pathway collection from EMP: The enzymes and metabolic pathways database. Nucleic Acids Res., (24):26–28, 1996.

Shaalan, K., Raza, H., (2009). NERA: Named Entity Recognition for Arabic, The Journal of the American Society for Information Science and Technology (JASIST), John Wiley & Sons, Inc., NJ, USA, 60(8): 1652–1663, July 2009.

Shaalan, K. and Raza, H. (2007). Person Name Entity Recognition for Arabic, Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Prague, Czech Republic, pp. 17-24, June.

Shamsi. F., Guessoum, A. (2006). A Hidden Markov Model –Based POS Tagger of Arabic, proceedings of the 8th International Conference on the Statistical Analysis of Textual Data, April 19 – 21, 2006, Besançon, France.

Somers, Harold L, Black B., Nivre J, Lager T., Multari A., Gilardoni L., Ellman J., & Rogers A. (1997). Multilingual Generation and Summarization of Job Adverts: The TREE Project accepted for Publication Fifth Conference on Applied Natural Language Processing, Washington DC USA.

Srihari, R. & Li, W. (2000). Information Extraction Supported Question Answering. Proceedings of the sixth conference on applied natural language processing. Seattle, Washington.

Sundheim, B. (1992). Overview of the Fourth Message Understanding Evaluation and Conference. Proceedings of the Fourth Message Understanding Conference (MUC-4), June, 1992, Morgan Kaufmann Publishers.

Sundheim, B., M., (1995). Overview of results of the MUC-6 evaluation. In Proceedings of the 6th Conference on Message understanding, November 06-08, 1995, Columbia, Maryland.

Sundheim, B. (1991). Overview of the Third Message Understanding Evaluation and Conference. In Proceedings of the Third Message Understanding Conference (MUC-3), May, 1991, Morgan Kaufmann, pp. 3-16.

Takeuchi, K. and Collier, N. (2002). Use of Support Vector Machines in Extended Named Entity Recognition, in proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002), Taipei, Taiwan, August.

Thompson, P. & Dozier C. (1997). Name Searching and Information Retrieval. In Proc. of Second Conference on Empirical Methods in Natural Language Processing.

Tomokiyo, L., Black, A., Lenzo, K. (2003): "Arabic in my hand: small-footprint synthesis of Egyptian Arabic", In EUROSPEECH-2003, 2049-2052.

Viterbi, A., .J. (1967).Error bounds for convolutional codes and an asymptotically optimal decodingalgorithm.IEEE Transactions on Information Processing, 13:260-269

Weischedel, R., Boisen, S., Bikel, D., Bobrow, R., Crystal, M., Ferbuson, W., Wechsler, A., & the PLUM Research Group. Progress in Information Extraction. Advances in Text Processing: Tipster Program Phase II. 1996.

Yu, S. Bai, S. & Wu, P. (1998). Description of the Kent Ridge Digital Labs System Used for MUC-7. In Proceedings of 7th Message Understanding Conference, USA (1998) pp: 51

Zhao, S. (2004). Information Extraction from Multiple Syntactic Sources. PhD thesis university of New York.

Zhou, G. 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), Philadelphia, July 2002, pp. 473-480.

Zitouni, I., Sorensen, J., Luo., X & Florian, R. (2005). The Impact of Morphological Stemming on Arabic Mention Detection and Coreference Resolution, In the Proceedings of the ACL workshop on Computational Approaches to Semitic Languages, 43rd Annual Meeting of the Association of Computational Linguistics (ACL05). June, Ann Arbor, Michigan, USA, pp. 63-70.