Abstract

This project, the Classification of content in Arabic Documents and the Extraction of Personal Information (CADEPI), has two objectives in presenting further study of the Arabic language in the national language processing (NLP) area. The first objective is the classification of Arabic contents using the Naïve Bayes classifier algorithm with five pre-processing features. These are general normalization, Arabic normalization, removing stopwords, light stemming and parts of speech. Comparisons between these pre-processing features were made to evaluate their performance with the NB classifier algorithm as singular and component features. Some of these features were used for the first time with the NB classifier algorithm such as choosing nouns and proper noun features using the parts of speech technique. The second objective is the automatic extraction of full personal names from Arabic text contents using the new Arabic person name dictionary (ANDic) which contains 82816 names. In addition a new experimental method (the reverses 10-fold cross validation method) was created during this project in order to evaluate a classification algorithm with a limited number of training set examples.

- *A Naïve Bayes text classifier for Arabic contents.*
- *Five pre-processing singular features for Arabic content classification tasks.*
- *Combinations of these pre-processing features as models for Arabic content classification tasks.*
- *A comparative study of singular and combinations of pre-processing features for Arabic content classification tasks, focusing on the Naïve Bayes classifier.*
- *The creation of a new method- the Reverse 10 crosses validation method - to evaluate text classification tasks using a limited number of data sets as training.*
- *Further study on Parts of speech tagging and the efficiency of light stemmer techniques in the Arabic language.*
- *A new dictionary of Arabic people names (ANDic) which contains around 82,816 Arabic names.*
- *An extraction of Arabic full names, focusing on the (ANDic) dictionary.*
- *Corpus of Arabic datasets and the new dictionary as an initial step towards standardising Arabic language work in the machine learning area.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Glossary

ANDic: *Arabic name dictionary*.

CADEPI: *Classification of content in Arabic Documents and Extraction of Personal Information.*

NB: *Naïve Bayes: a machine learning classifier algorithm.*

Pre-processing features: *These procedures apply before the NB classifier as data preparation process.*

Singular features: *Stand alone features- namely using one pre-processing feature with NB only.*

Combination of features: *group of features working together.*

BL: *Baseline model which uses NB only during the classification task.*

G1: *General normalization which is a pre-processing feature to convert the word to the normal form.*

G2: *Arabic normalization which is a pre-processing feature to convert some letters into a unifier form.*

ST: *Removing stopword feature.*

LS: *Light stemmers which remove very common prefixes and suffixes from Arabic words.*

NP: *Choosing nouns and proper nouns using the parts of speech technique.*

Dic: *Using only the ANDic dictionary as a model.*

Dic&NP: *Using the ANDic dictionary and choosing nouns and proper nouns using the parts of speech technique as a model.*

Dic&NPJ: *Using the ANDic dictionary and choosing nouns, proper nouns and adjectives using the parts of speech technique as a model.*

1. Chapter 1: Introduction and Context

1.1. Introduction

The number of written documents has grown dramatically thanks to online web applications and services that have produced a huge number of easily available electronic libraries. Therefore, manual indexing, classification, information retrieval and extraction are important for gaining information from text contents (documents) but have become difficult and very time consuming. For that reason machine learning and data mining tasks have become important fields to help users deal with these functions easily.

The number of Arabic language documents has grown as it has for all human languages. Despite this growth in the number of Arabic texts, there is a lack of automatic data mining tools, which can be applied to this language using machine learning methods (Maamouri et al., 2004). Therefore, this thesis provides further study of this language by focusing on machine learning tasks (classification of contents and extraction of personal information).

1.2. Aims and Objectives

This project – the Classification of the content of Arabic Documents and the Extraction of Personal Information (CADEPI) – aims to employ machine learning (ML) and data mining (DM) techniques to Arabic documents with two objectives. The first objective is to classify the content of Arabic documents to assist users to easily determine the general subject or overview topic of the document. The second objective is the efficient extraction of personal information, such as the name and phone number from the Arabic text; this extraction supports the user's focus on the important information inside the context.

1.3. Research Contributions

This project contributes the following to the field of Arabic machine learning tasks.

It presents:

- A Naïve Bayes text classifier for Arabic contents.
- Five pre-processing singular features for Arabic content classification tasks.

- Combinations of these pre-processing features as models for Arabic content classification tasks.

- A comparative study of singular and combinations of pre-processing features for Arabic content classification tasks, focusing on the Naïve Bayes classifier.
- The creation of a new method- the *Reverse 10 crosses validation* method - to evaluate text classification tasks using a limited number of data sets as training.
- Further study on Parts of speech tagging and the efficiency of light stemmer techniques in the Arabic language.
- A new dictionary of Arabic people names (ANDic) which contains around 82,816 Arabic names.
- An extraction of Arabic full names, focusing on the (ANDic) dictionary.
- Corpus of Arabic datasets and the new dictionary as an initial step towards standardising Arabic language work in the machine learning area.

1.4. Organisation of this dissertation

This thesis comprises seven chapters briefly described as follows:

- Chapter 2 briefly presents an introduction to the Arabic language structure in order to illustrate general overviews of Arabic language characteristics and challenges.

- Chapter 3 describes previous work which has been completed in three fields. The first is the classification content of Arabic documents. The second is the pre-processing of Arabic words and the third is Arabic name extraction.

- Chapter 4 describes the CADEPI project design and its processes in order to show all the project’s models which are used during the project’s experiments.

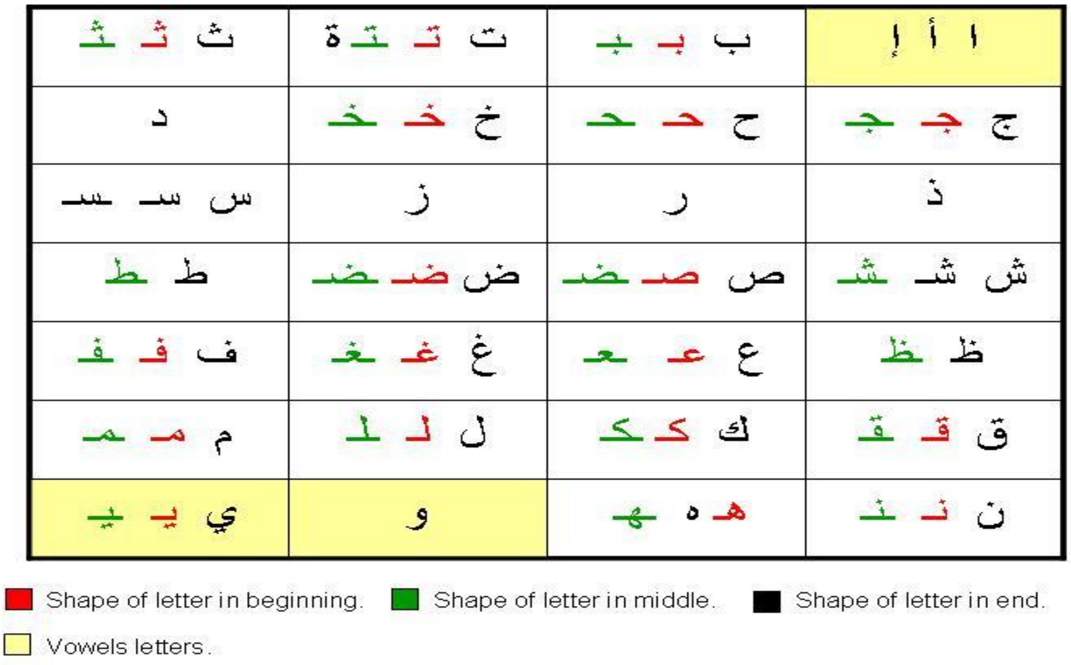
- Chapter 5 explains the results of the project’s experiments by focusing on the most preferable algorithm for the classification of Arabic contents of documents and personal name extraction.

- Chapter 6 presents the conclusion to the CADEPI project highlighting important results of the project, future work and future project phases.

2. Chapter 2: Arabic language structures

2.1. Introduction

Arabic is the mother language of more than 246 million people; it is also one of the six official languages of the United Nations (UN). The Arabic alphabet consists of 28 letters; three letters are vowels (ي , و ,ا), and the rest are consonants (Aljlayl & Frieder, 2002). The direction of writing in Arabic is from right to left. Arabic words consist of a number of related and unrelated letters. A letter can have four different shapes: isolated, at the beginning, in the middle, or at the end. The shape depends on its location in the word. Hence, the letter, which has different forms, is a related letter see Figure (1, 2).



ث ثـ ـثـ	ت تـ ـتـ ة	ب بـ ـبـ	ا أ إ
د	خ خـ ـخـ	ح حـ ـحـ	ج جـ ـجـ
س سـ ـسـ	ز	ر	ذ
ط ـطـ	ض ضـ ـضـ	ص صـ ـصـ	ش شـ ـشـ
ف فـ ـفـ	غ غـ ـغـ	ع عـ ـعـ	ظ ـظـ
م مـ ـمـ	ل لـ ـلـ	ك كـ ـكـ	ق قـ ـقـ
ي يـ ـيـ	و	هـ ه ـهـ	ن نـ ـنـ

🟥 Shape of letter in beginning. 🟩 Shape of letter in middle. ⬛ Shape of letter in end.

🟨 Vowels letters.

Figure 1: The Arabic alphabet and the shape of their letters at the beginning, in the middle and at the end.

The Arabic language has a very wide range of characters that refer to two genders (i.e. feminine and masculine); and three numbers (i.e. singular, dual, and plural words). Arabic words are classified into three main parts of speech: nouns (including adjectives and adverbs), verbs, and particles. Moreover, Arabic words have three grammatical cases: nominative, accusative, and genitive. A noun has the nominative case when it is the subject;

the accusative when it is the object of a verb; and the genitive when it is the object of a preposition (Al-Harbi et al. 2008).

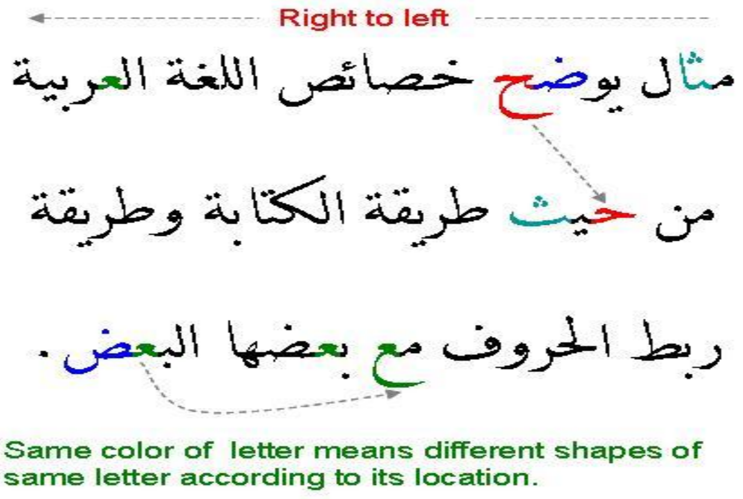

Right to left
مثال يوضح خصائص اللغة العربية
من حيث طريقة الكتابة وطريقة
ربط الحروف مع بعضها البعض.
Same color of letter means different shapes of same letter according to its location.

Figure 2: A letter with the same colour means different shapes of the same letter according to its location.

2.2. The Arabic language morphology and diacritics

Arabic is a semantic language with a composite morphology of diacritics, which are symbols above or below letters used in orthography and as a sign for its grammatical case (Syiam et al., 2006). Arabic syntax uses these symbols to change the letter form in order to change its pronunciation. Arabic words have different sounds according to their position within the sentence. Figure (3) shows three important signs: “dumah,” “ksrah” and “fatha” (from left to right) and their effect on the letter sound. Although these characteristics reflect flexibility and power, they also indicate the difficulty of processing Arabic words electronically during the pre-processing stage.

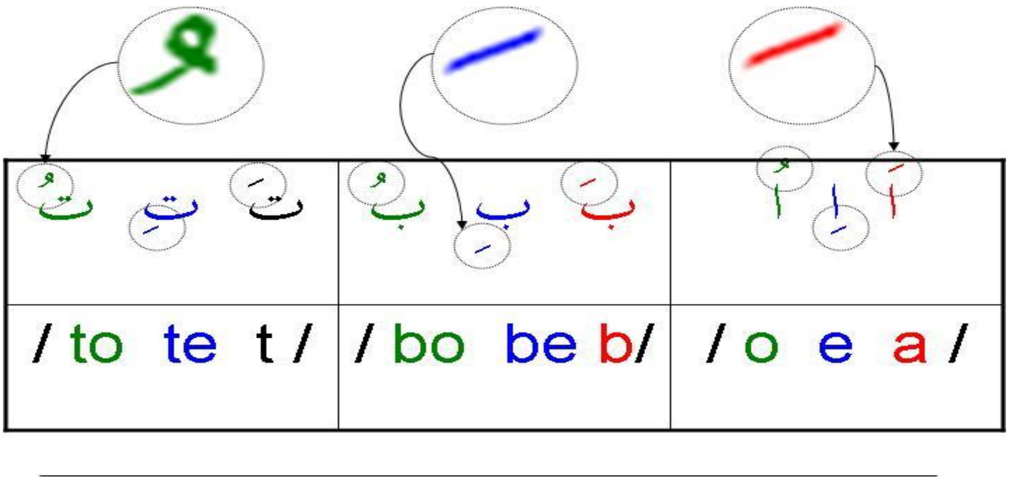
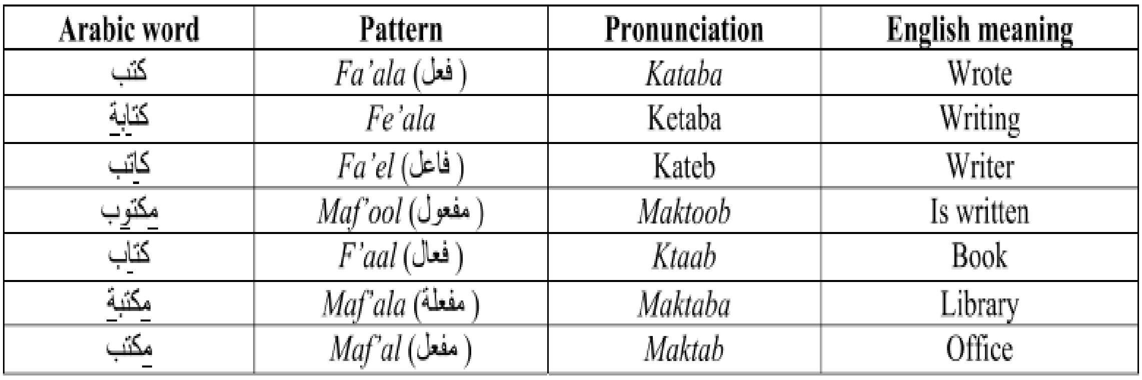


Figure 3: Arabic syntax signs “dumah,” “ksrah” and “fatha” (from left to right).

2.3. Arabic word patterns

The Arabic language includes a number of patterns as semantic templates. They are useful to classify Arabic sentences into their parts of speech. For example the pattern Faa'el (فاعل) represents the subject of the verb, the pattern maf'ool (مفعول) indicates the object of the verb. The pattern Fa'ala (فعل) is a very common one because the majority of Arabic words are derived from this pattern. Table (1) explains an example of the root word kataba (كتب) (which in English means 'book') in order to show the effect of these different derivations on the word meaning. The letters that have been added to the main root of the word are underlined (Syiam et al., 2006).

Table 1: different derivations of the root word "kataba" (Syiam et al. 2006)



Arabic word	Pattern	Pronunciation	English meaning
كتب	*Fa'ala* (فعل)	*Kataba*	Wrote
كتابة	*Fe'ala*	Ketaba	Writing
كاتب	*Fa'el* (فاعل)	Kateb	Writer
مكتوب	*Maf'ool* (مفعول)	*Maktoob*	Is written
كتاب	*F'aal* (فعال)	*Ktaab*	Book
مكتبة	*Maf'ala* (مفعلة)	*Maktaba*	Library
مكتب	*Maf'al* (مفعل)	*Maktab*	Office

2.4. The Arabic language typography and ambiguity

Typographic variance, in the Arabic language, is supported widely by diacritic signs and variants of Arabic language accents; thus, the Arabic language is highly ambiguous. More explanation is provided below using demonstrative examples:

1. With the omission of short vowels, the word كاتب (writer) for example, can represent the following seven word forms:

Table 2: Diacritic sign effects (omission of short vowels) in an Arabic word

Form	كَاتِبٌ	كَاتِبً	كَاتِبٍ	كَاتِب	كَاتَبَ	كَاتَب	كَاتب
Pronunciation	Kaatibun	kaatiban	Kaatibin	Kaatib	Kaataba	Kaatab	Kaatb

The Arabic language has a strong syntax to organise the pronunciation of Arabic words according to their position in a sentence; these forms in Table (2) reflect different pronunciations for the word depending on its position within the sentence. However, all of these word forms have the same meaning, which is writer in English and kaatib in Arabic.

2. The long /aa/ in Arabic known as (Maad ~) especially at the end of words, can be represented by various letters, such as (ا،ى،آ،ه،ة). For example:

Table 3: The long /aa/ (Maad ~) effects in an Arabic word

English word	Alexandria	Asia	Syria
Arabic word form 1 Pronunciation	الإسكندريا /alaskndria/	آسيا /asia/	سوريا /soriaa/
Arabic word form 2 Pronunciation	الاسكندريه /alaskndrih/	آسيه /asiah/	سوريه /soriah/
Arabic word form 3 Pronunciation	الاسكندرية /alaskndrih/	آسية /asiah/	سورية /soriah/

3. Alternation between ي (yaa') and ى ('alif maqSuura) is a very common mistake in the Arabic script. Thus, it is not strange to find the same word written in two forms in the same script especially with ي (yaa') and ى ('alif maqSuura). For example, /Mustafa/ an Arabic name:

Table 4: Alternation between ي (yaa') and ى effects in an Arabic word

مصطفى /Mustafa/	مصطفي /Mustafi/

4. The rules for determining the hamza position are notoriously complex. For example, Ahmed is a common name in Arabic; it can be represented by two forms according to ء/hamza/.

Table 5: Effects of the Hamza position in an Arabic word

أحمد with hamza	احمد without hamza

5. Alternation between ه (haa') and ة (taa' marbuuTa). For example, this alternation occurs in names like Audah being spelled as follows:

Table 6: Alternation between ه (haa') and ة (taa' marbuuTa) effects in an Arabic word.

عوده /Audah /	عودة / Audah /

6. Compound names are often written as either one word or two. For example, /`abd-ullah / is written as the following:

Table 7: Effects of compound names in an Arabic word

عبد الله /Abdu-llah / with space	عبدالله / Abdullah / without space

This type of Arabic name is very common; there are no specific rules as to whether to write them with spaces or without. Therefore, two forms with the same meaning are acceptable (Halpern, 2009)

2.5. Summary

The Arabic language is considerably complex and is consequently a challenge to Natural Language Processing (NLP) for because the Arabic word structure depends on joined letter forms, i.e. different forms of the Arabic letter, according to its position in the word. The Arabic word has two genders (i.e. feminine and masculine); and three numbers (i.e. singular, dual, and plural words) which means there are different forms. There are no special characteristics to determine any part of speech in an Arabic sentence such as the first capital letter in proper noun words in the English language; all parts of speech have the same characteristics.

3. Chapter 3: Background and previous work

3.1. Introduction

This chapter focuses on previous work in the classification of the contents of Arabic documents and Arabic personal name extractions and which are a part of the CADEPI project objectives. Also, it shows related Arabic word processing works such as word normalization, part of speech techniques, Arabic stopwords and Arabic word stemming. Reviewing this knowledge is supportive in building strong overviews about project goals.

3.2. Classification of the content of Arabic Documents

Text classification techniques for English, French, German, Spanish, Chinese, and Japanese documents have been investigated and explored in several research projects. However, in the Arabic language there is little ongoing research in automatic Arabic document classification (Al-Harbi et al. 2008). Also, on the commercial side there is only one automatic Arabic document categorization referred to as *Sakher auto* categorizer (Sakhr, 2010)

There have been two main strategies in the constriction of the content classification system. The first strategy is the rule-based approach which uses background knowledge – e experience – to write some rules for content classification. The second strategy is the learning-based approach which uses manual classified texts as examples to create rules automatically for content classification (Lewis, 1994).

Therefore, establishing a classification system in any language through machine learning or learning -based methods consists of the three following phases:

1. Collect and label the text documents in corpora.
2. Select a set of features to represent defined classes.
3. The appropriate classification algorithms chosen must be tried and tested using the collected corpora in the first stage.

Therefore, these phases are convenient to the Arabic language classification process, and they can be considered as main stages in this project as well. See Figure (4).

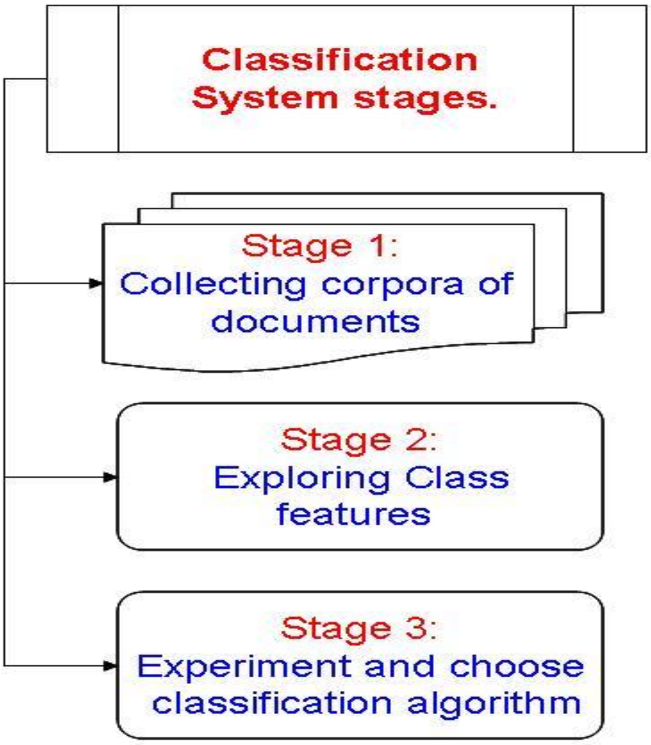

Classification System stages.
Stage 1: Collecting corpora of documents
Stage 2: Exploring Class features
Stage 3: Experiment and choose classification algorithm

Figure 4: Three main stages of the classification system

3.3. Previous work with Arabic documents

This section has been divided into two parts. The first part relates to previous work that has been performed with Arabic document classifications. The second part relates to previous work that has been completed in Arabic word processing such as work with normalization, stopwords and stemming.

3.3.1. Classifications

Previously published Arabic document classification research has revealed substantial results through different machine learning algorithms and different techniques. The following section will present some of this research by focusing on their results and techniques.

Al-Harbi et al (2008) experimented with the SVM algorithm and Clementine for the C5.0 decision tree algorithm by RapidMiner and Clementine (Data mining tools). Their aim was to evaluate their performance regarding the classifying of Arabic texts which they collected from different sources (e.g. Saudi Press Agency (SPA), Saudi News Papers (SNP), web sites, discussion forums, Islamic topics and Arabic poems). They were also categorized into many classes such as cultural news, sports news, social news, economic news, political news, general news, IT news, Islamic topics and Arabic poems. Their overall accuracy average was 68.65% by SVM and 78.42% by C5.0 (Al-Harbi et al., 2008).

El-Halees applied the maximum entropy method to classify Arabic documents that were collected from the Aljazeera Arabic news channel. The documents were categorized into six domains: politics, sports, culture and arts, science and technology, economy, and health. The maximum entropy method was applied with and without pre-processing stages. Without any pre-processing stages the accuracy was 68.1%; with only normalization in the pre-processing, the performance increased to 70.25%. Performance increases to 71.20% by applying the normalization and tokenizing pre-processing stage together. Finally, by using parts of speech in an experiment only nouns and proper nouns were used; other words in the text were excluded. In this case, the performance increased to 80.41%, which was the largest increase (El-halees 2007). El-Halees (2006) described a method based on association rules to classify Arabic documents. The classification accuracy reported was 74.41% (El-halees, 2006).

El-Kourdi et al used the Naïve Bayes (NB) algorithm to automatically classify Arabic documents. The training set, which was used in the classification experiment, was collected from the website www.aljazeera.net, which is the website of Aljazeera (the Qatari television news channel in Arabic). The 300 web documents corresponding to the five categories used for this work are sports, business, culture and art, science, and health. The average accuracy of overall categories was about 68.78% (Elkourdi et al., 2004).

Sawaf et al used statistical classification methods such as maximum entropy to classify and cluster Arabic NEWSWIRE articles. The articles covered politics, economy, culture and sports. The best classification accuracy they reported was 62.7% (Sawaf et al., 2001).

The K-nearest neighbours (KNN) classifier was used to classify the data set, which consisted of 15,000 collected Arabic text documents from Internet sites. They were categorized into three classes: politics, sports and economics. Two approaches were used in the pre-processing stage. The first approach was the stemming approach, which found the three-letter roots for Arabic words without depending on any root or pattern files. The second approach was light stemming, which removes the frequent suffixes and prefixes from the words. The experiments showed that light stemming outperformed the stemming approach because stemming affects the meanings of words (Al-refai et al., 2007).

During the revision of the previous work, the main difference between the classifier algorithms was not clear except for the difference in the simplicity of their implementation.

Some algorithms can be customized such as by adding some bias or weight for a particular property value (e.g., Naïve Bayes) whilst others cannot be customized easily (e.g. support vector machine).

One of the most important techniques is 'light' stemming, which was confirmed by the experiment; it outperforms the stemming approach because stemming affects the word meanings (Al-refai et al., 2007). In addition, the parts of speech technique is used in the pre-processing stage to implement the classification algorithm on the Arabic noun and proper noun only. It is confirmed by experiment and attained a high score in classification accuracy.

3.3.2. Arabic word processing

The effort which was made in Arabic classification contents is undoubtedly very useful for any future work; it focused on some important steps and techniques. One important factor is to consider the pre-processing contents stage. This stage occurs before implementation of the classifier algorithm to prepare and filter raw data. In addition, different forms for the same word and semantically repeated words are reduced in this stage to provide a significant domain, enabling efficient classification. The section below shows some important techniques that have been used in this stage.

Al-Shalabi et al's morphology system was developed using different algorithms to find the roots and patterns. This system's goal was to extract the root by removing the longest possible prefix. Subsequently, the first five letters of the word were checked (Al-shalabi & Evens, 1998).

A slightly different methodology involves removing the prefixes and suffixes algorithm from Arabic words to match the remaining word against the patterns of the same length to extract its root. However, it is important to check that it is not removing part of the root as part of the word's prefixes or suffixes is time-consuming (Larkey et al., 2002).

The majority of Arabic words are derived from a tri-lateral root that involves between 80 and 85% of Arabic words. The remaining words have either a quad-letter root, penta-letter root or hexa-letter root. Thus, Al-Fedaghi and Al-Anzi's algorithm tried to find the root of the word by matching the word with all possible patterns with all possible affixes attached to it (Khreisat, 2009).

The root extraction technique has been developed by El Kourdi, and involves transferring all Arabic word derivatives to their original form or single common root. This method benefits in terms of reducing the domain structure and supporting semantic word relationships. As a result, this will decrease repeated words semantically (Elkourdi et al., 2004).

El-halees has employed a normalization pre-processing technique using two steps. In the first step, the script is converted to UTF-8 encoded, and punctuation and non-letters are removed. In the second step, some Arabic letters are normalized such as إ ,أ, and آ and are converted to normal ا , and ى is replaced by ي and ة to ه. Therefore, the domain of the script word is reduced to increase the accuracy of classification (El-halees, 2007).

Aljlayl and Frieder have developed light stemming to remove some defined prefixes and suffixes from the word instead of extracting the original root. The main idea for using light stemming is that root extraction algorithms sometimes affect the meanings of words; thus removing clear prefixes and suffixes is very efficient in comparison to the extraction of the word's root (Al-refai et al., 2007).

3.3.3. The Naïve Bayes classifier algorithm

The Naïve Bayes algorithm is one of the most popular algorithms in machine learning because it is highly efficient especially in classification tasks. In addition it is easy when customizing to add some bias such as to add weight to important words in specific classes. Therefore, this section will illustrate this algorithm:

Let D be a document representing a set of finite words D = {w1, w2,..., wn}.

Let docsi be the number of documents in category Ci.

Let |Examples| be the number of documents in the training set of labelled documents.

ni is the total number of distinct terms in all training documents labelled Ci.

Texti is where Nk,i is the number of times wk occurs in Ci, a single document generated by concentrating all the training documents for category Ci.

Step 1: Collect the vocabulary, which is defined as the set of distinct words in the whole training set.

Step2: For each category Ci, Cj etc, do the following Compute P(Ci) = |docsi|/|Examples|.

Step3: For each word wk in Vocabulary Compute $P(wk/Ci)= (Nk,i +1)/(ni +| Texti |)$

Step3: $P(w1,...,wn|Cj)=P(w1|Cj)*P(w2|Cj)*...*P(wn|Cj)$ *(5)*

(Elkourdi et al., 2004).

3.3.4. Evaluation

The previous work that focused on the Arabic document classification problem was useful in two ways. The first is that the pre-processing stage to prepare and normalize the data set was an extremely important stage that had a significant effect on the classification performance. Therefore, qualified work in this stage will support classifier algorithms in their aim to achieve a substantial result. The second view is that popular classifier algorithms such as support vector machine (SVM) and Naïve Bayes are used in Arabic documents without significant differences between them. Thus, most classifier algorithms have proximate efficiency to classify Arabic documents.

However, there is no standard corpus of Arabic documents available online for the comparison task between different classification techniques which have been used to verify and compare fairly between their results and performance. Generally, the lack of standardization makes this research field less organized on a global scale.

3.4. Extraction of Personal Information

Natural Language Processing (NLP) refers to the recognition and classification of proper names in text; for example, persons, locations and names of organizations. Various types of significant NLP applications exist such as Information Extraction, Information Retrieval and Machine Translation. Moreover, the important information in a text is normally located around proper names, thus determining them is a very useful task which highlight crucial information inside the script (Shaalan & Raza, 2007).

The name identification task has been conducted quite intensively over the past few years. Many researchers have studied this problem over a range of languages; however, only a few researchers have focused on Arabic text (Shaalan & Raza, 2007). One of the causes may be that identifying proper names is not easy from a practical point of view; a name in Arabic does not have particular properties (e.g. beginning with a capital letter as in English) to identify it easily within the document contents. Therefore, research is required to find the diagnostic method which can be used to extract names efficiently.

3.4.1. Arabic name components and properties

This section will study the Arabic name components intensively because it might be very helpful to find some features to extract personal names efficiently.

According to Ibn Auda (2003), the Arabic name fundamentals may be divided into five main categories:

1. An ism (pronounced ISM), a personal, proper name given shortly after birth. Such names are Ahmad [Ahmed], Mohammad [Mohammed], Musa [Moses] and Ibrahim [Abraham].

2. A kunya (pronounced COONYAH), a surname, as the father or mother of someone; for example, abu Da'ud [the father of David], umm Ahmed [the mother of Ahmed]. It is meant as a prefix of respect or reverence. Generally, in the Arabic language married persons are simply called by their kunya (especially) (abu for a married man, or umm for married ladies, + the name of their first-born child). When using a person's full name, the kunya comes before the personal (given) name: Abu Yusuf Hasan [the father of Joseph, Hasan], Umm Ja'far Fatima [the mother of Ja'far, Fatima]. For further explanation see figure 5.

3. A nasab (pronounced NAHSAHB), a pedigree, as the son or daughter of someone; for example, ibn 'Umar [the son of Omar], bint Saad [the daughter of Saad]. The nasab terms (ibn or bint) follows the ism in practice: Hasan ibn Fahad [Hasan the son of Fahad], Sumayya bint Ali [Sumayya the daughter of Ali]. See figure 5. Many historical personages are better known to us by their nasab than by their ism or name. For example, the historian ibn Khaldun, the traveller ibn Battuta, and the philosopher ibn Sina [Avicenna]. Nasabs may be extended for a number of generations, as may be noted in the example below containing two generations of nasab: Abu al-Qasim Mansur ibn al-Zabriqan ibn Salamah al-Namari.

4. A laqab (pronounced LAHKAHB), a mixture of words made into an epithet, usually religious, relating to nature, a descriptive, or of some marvellous quality the person had. For example, al-Fadl [the Prominent]. Laqabs go behind the ism such as Harun al-Rashid [Aaron the Rightly guided].

5. A nisba (pronounced NISSBAH), a name derived from a person's trade or profession, place of residence or birth and religion; for example, al-Hallaj [the dresser of cotton], Al Msri [The Egyptian], Islami [Is- lamic]. Generally, nisbas

follow the ism or a nasab such as Ahmed ibn Ibrahim Alamri see Figure (5) (Shaalan & Raza, 2007).

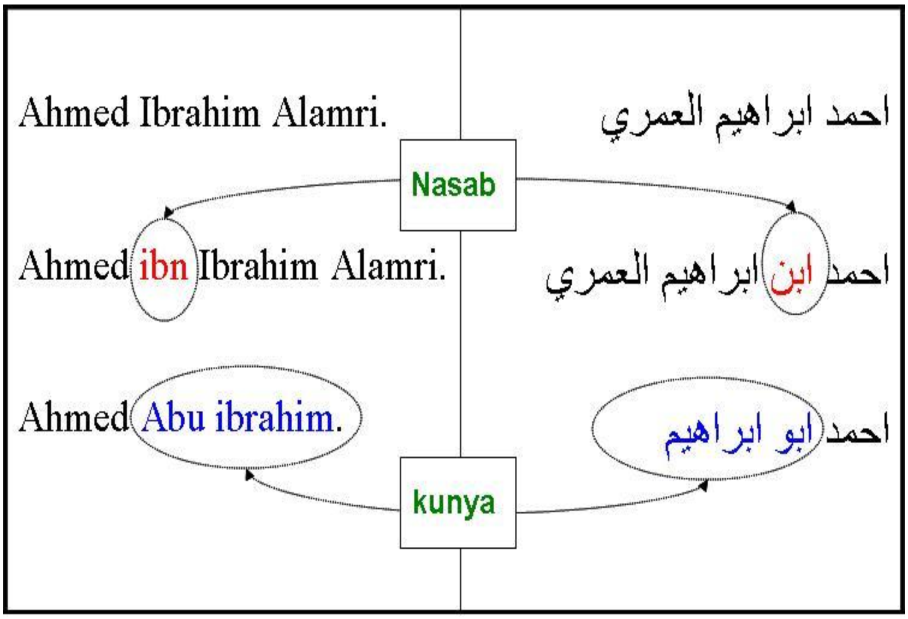


Figure 5: Name component examples.

3.5. Previous studies in Arabic name extraction

Only a few Arabic name extraction research studies have been published so far; however, these studies provide important information, and consequently the following section will discuss some of this research by focusing on their results and techniques. This section has been divided into two parts. The first part relates to previous work that has been studied in Arabic name extraction. The second part relates to Arabic dictionaries by focusing on Arabic name dictionaries.

3.5.1. Name extraction

Abuleil (2002) developed name extraction techniques by creating a set of rules to predict where the names are positioned in the text. These rules are based on two schemes: the keyword and some special verbs. Mainly, names seem to appear close to one of these keywords such as manager, professor, president and country or special verbs such as “said” and announced in Arabic text. Then the phrase of the name is extracted by assuming some regulation such as the name should not be more than three words away from the keyword or the special verb. Moreover, the longest name is 7 words. Next, the lexicon is used to determine the names within the phrase of a name. Finally, the research classifies each of them and saves them in the name database (Abuleil, 2002).

KHOJA has developed an Arabic part of speech tagger (APT) that uses statistical and rule-based techniques and also tagsets that have been derived from traditional Arabic grammatical theory (Khoja, 2001).

Halpern (2009) has developed recognition of Arabic named entities through a lexicon-driven approach. It exploits a large database of Arabic personal names (DAN), which currently contain over five million entries. Combining a lexicon-driven approach with statistical methods is the key to achieving effective processing of Arabic names (Halpern, 2009).

Abuleil (2004) developed a technique to extract proper names from texts to build a database of names. This work was done using three central stages: 1) marking the phrases that might include proper names, 2) constructing graphs to represent the words in these phrases and the relationships between them, and 3) applying rules to produce the names, classify them and save each of them on a database (Abuleil, 2004).

A list of the trigger words technique, which indicates neighbouring words are probably located close to personal names, and have been used because there are some verbs and adjectives which mainly surround personal names (Mesfar, 2007)

AlGahtani and McNaught (2009) developed an Arabic named entity recognition system based on Maximum Entropy modelling using features such as Lexical, POS tagging, Trigger, Class of previous word and the Global feature. It was mixture between the Rule-based and the Corpus-based approaches. Their system achieved a 91.7 F-measures value (Algahtani & McNaught, 2009)

Elsebai et al (2009) developed and implemented a personal name entity recognition system for the Arabic Language dependent on a rule based approach with a set of keywords, which are a guide to the phrases that may contain personal names. They did not use any predefined personal name such as in a dictionary; their system achieved an 89% F-measure value (Elsebai, 2009).

The Arabic part of speech tagging system was developed using the Transformation-Based Learning (TBL) method for the modern standard Arabic text. It has two templates rules: lexicalized and non-lexicalized. The non-lexicalized rule depends only on surrounding tagging information and the lexicalized one depends on the dictionaries. This tagger system achieved an accuracy rate of 96.9% (AlGahtani et al., 2007).

The AGARAB system has been developed by Maloney et al (1988). It consists of two major components: a morphological tokenizer and a name finder. The morphological tokenizer aims to normalize word levels to perform lexical scanning efficiently. A Name Finder unit, which identifies names and other extraction targets by using word lists and pattern action rules annotates the text with appropriate tags for each extracted item in Figure (6) (Maloney et al., 1988).

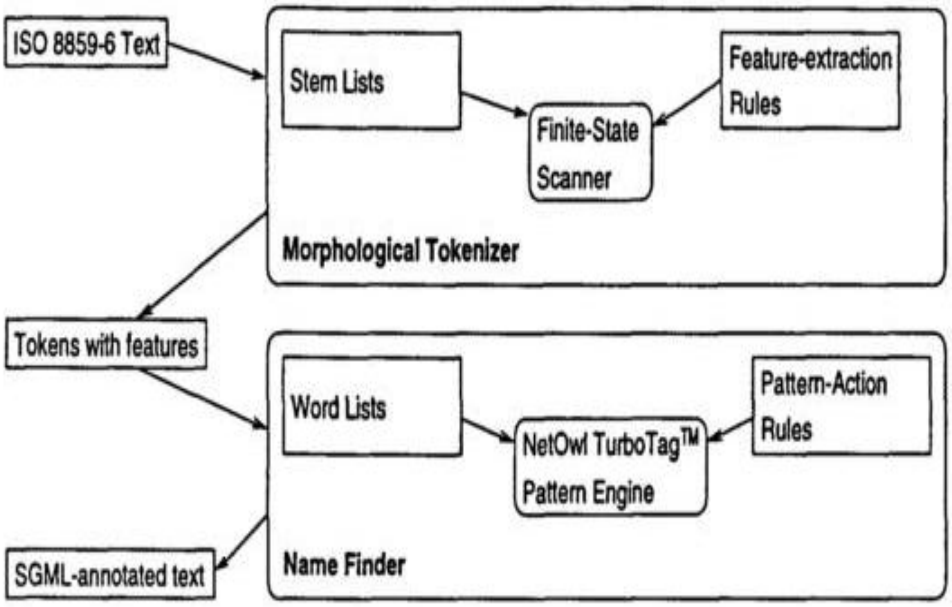


Figure 6: TAGARAB system architecture. (Maloney et al. 1988)

3.5.2. Dictionary

A dictionary is considered to be the essential part of any natural language application because it is essentially needed for parsing, parts of speech tagging, text generation, information extraction, and information retrieval systems. All of these applications or others in the NL area (without a high-quality dictionary) did not work efficiently (Abuleil & Evens, 1999).

Gazetteers dictionaries are derived using the data collected from different sources such as the Treebank corpus and government databases, which, according to Shaalan et al, contain a total of 472617 entries. These entries consist of different types of data such as (first, second, third) person names, full names, person titles, nicknames and job names. This dictionary was used in name recognizing; the average precision and recall achieved was 85.5% and 89%, respectively (Shaalan & Raza, 2007).

A system for building an Automatic Lexicon for the Arabic language was designed by Al-Shalabi and Kanaan. This lexicon includes words with some related information such as word roots, patterns, parts-of-speech tags (noun, verb or particle) and lexical attributes (gender, number and case). They are stored with the word in the lexicon. This system has been tested using some Arabic text documents which were taken from the holy Qur'an and the Saudi Arabian National Computer Conference abstracts. It achieved an accuracy of about 96% (Al-shalabi & Kanaan, 2004).

3.5.3. Evaluation

In previous research into Arabic name extractions, different techniques were used such as dictionaries, name tagging, and grammar rules. These techniques contributed efficiently to name extraction. However, standard techniques, such as using the same edition of the Arabic name dictionary, Arabic name tagging tools and inclusive grammar rules — which are extremely convenient for extracting Arabic names — are not emphasised clearly. Furthermore, a collection of Arabic names or Arabic name dictionaries must be studied carefully as specific types of dictionaries in order to identify their features and utilise them during any NLP application generally and the extraction process especially.

Using standard tools and used confidently, will provide a solid basis for all researchers and developers to measure and evaluate their techniques' performances against previous work's performances. Nevertheless, that does not undervalue the previous work that has been completed regarding Arabic name extraction. That work has provided a wide range of advice about Arabic name components and complexity.

3.6. Performance measures

Practical evaluation of the accuracy of theories is essential to machine learning research in order to measure their results statistically (Mitchell 1997). These results often require standard and confident evidence to determine their progress and prove their performance appreciably against that of previous work. Therefore, recall, precision, f-measures and t-tests are described in the next sections as they are used in the CADEPI project as performance measures.

3.6.1. Recall, Precision and the F-measure

Precision and recall are two commonly used statistical measures in machine learning and data mining tasks. Before, defining each one confusion Matrix and some abbreviations should be explained in Table (8):

Table 8: Confusion Matrix (Sokolova & Lapalme, 2009)

		Actual	
		Positive	Negative
predicted	Positive	TP	FP
	Negative	FN	TN

- *TP: number of predicted objects and labelled as belonging to the relevant class.*
- *TN: number of unpredicted objects and labelled as belonging to the irrelevant class.*
- *FP: number of predicted objects and labelled as belonging to the irrelevant class.*
- *FN: number of unpredicted objects and labelled as belonging to the relevant class.*

Recall is a measure of comprehensiveness and is defined as the percentage of correct assignments (TP) divided by the total number of correct assignments (TP + FN) in the following form:

$$Recall = TP/ (TP + FN).$$

Precision can be seen as a measure of loyalty and it is defined as the percentage of correct assignments divided by the total number of all assignments in the following form:

$$Precision = TP/ (TP + FP).$$

The F-measure is a combination of recall and precision with an equal weighting in the following form:

$$F\text{-}measure = 2(Recall{*}\ Precision)/ (Recall+ Precision).$$

(Yang & Liu, 1999; Davis & Goadrich, 2006)

3.6.2. The T-test

The T-test is a very common statistical test that is used to find out if there is a real difference between the means (averages) of two different methods. It is frequently used to observe the difference between two classifier algorithms and to monitor the difference between two models of classification such as the baseline algorithm alone and the baseline algorithm with extra feature accuracies. The T test value (p value) shows that if the p value is less than the threshold, which is usually (0.10, 0.05, or 0.01), it indicates that the difference between the two models is significant and hence one model is better than the other (Soon et al. 2001).

3.7. Summary

To summarize the previous work, an effort has been made when classifying Arabic documents and Arabic name extractions to create very useful guidelines for any future work; this work highlighted some important techniques such as light stemming and parts of speech and their performance that are absolutely essential in the Arabic document pre-processing stages in both content classification tasks and in name extraction tasks. Therefore, some of these techniques have been applied in this project as models with machine learning algorithms as will be seen in the next chapter.

However, a general weak point is attributed to this effort. There are no standard and shared tools between NLP researchers in the Arabic language such as a corpus of Arabic documents which includes the common Arabic subjects, parts of speech technique, Arabic word stemming technique and highly competent Arabic dictionary in order to accurately support comparisons and competitive purposes between different Arabic NLP studies.

4. Chapter 4: Project Design

4.1. Introduction

In the CADEPI project, some features such as General Normalization, Arabic Normalization, Removing stopwords, light stemmers and the parts of speech technique are included in the Naïve Bayes classifier algorithm used to classify the contents of Arabic documents. Assessing the performances of these features might lead to discovering more efficient features which could be recommended as ideal models of the Arabic contents classification task with the basic machine learning algorithm. Furthermore, in this project, extracting the Arabic personal names in the Arabic text contents is investigated by using a new dictionary (ANDic). It contains 82816 Arabic names; I am not aware that it has been used in data mining before.

The particular design of this project includes details of technical requirements, developmental steps, experiment plans and measurement methods as described in the following sections.

4.2. Classification categories

As with all human languages, Arabic documents contain a variety of subjects that encompass every area such as history, health, science, economy, and politics. Thus, the classification of the contents of an Arabic document in this project cannot cover all its categories. However, classification will be made with reference to three classes: economy, politics, and sport.

Substantial reasons support the choice of economy, politics and sport as the first group of Arabic content categories. The first reason is that the media publishes reports on these subjects frequently because they are of considerable human interest. The second reason is that these categories have widely intersecting word domains with other Arabic subjects. This means that just on the basis of words used itself, classifying them efficiently is truly a challenge.

4.3. Data set collection

English text classification tasks universally use the Reuters data set collection as a standard evaluation corpus which has different versions such as the Reuters-215782 collection which is one of the most frequently used versions. This standard set supports researchers evaluating and comparing their results with other results; in contrast the lack of a publicly available Arabic corpus for weighing up text classification algorithms was one difficulty experienced at the beginning of the CADEPI project (Al-shalabi et al., 2006).

Since there is no publicly available Arabic text classification corpus which could be used as a training set to investigate the effectiveness of our model text classifier, a new corpus was collected from particular websites which belonged to three different categories. Gathering this training set involved three steps as follows:

1. The first step involved studying the characteristics of these three categories (politics, economics and sport) to obtain important information such as key words, features, important published sources of these categories and general information that could increase the quality of the training set.

2. The second step is to use this information (characteristics of categories) as a guideline to collect a significant number of documents published on respectable websites like official institutes or government departments, specialist newspapers and magazines in economics, politics and sport. Examples of organizations include the Saudi Arabia Monitory Agency (SAMA), the institute of banking in Saudi Arabia as well as the Al-Riyadh and Al-Eqtsadiah newspapers.

3. The third step involves the collaborative data set which is collected from different sources in order to test the CADEPI project in the real world with unbiased test data.

Therefore, 450 Arabic documents were collected from different sources as follows: 150 documents on the Economy, 150 documents on Politics and 150 documents on sport. In this Arabic dataset, each document was saved in a separate file and labelled within the main directory for all categories as a main folder of the training set. These data were refined many times so as to choose documents that were unambiguously related to classification classes in order to prepare highly precise training sets from each class to classify the future document accurately. Table (9) shows the number of documents and sources in each category.

Table 9: Data Set Sources

Sources	Economic	Political	Sport
Saudi Arabia Monitory Agency (SAMA)	5	-	-
Al-Riyadh newspapers	15	23	50
Al-Eqtsadiah newspaper	55	10	-
Capital Market Authority (CMA)	5	-	-
Al-ahram newspaper	5	7	-
Al-jazerah.net TV	30	50	50
Middle East newspaper	25	30	-
Al-Watan newspaper	10	30	50
Total	150	150	150

Furthermore, the collaborative dataset was gathered from different sources as a test dataset. These sources are not included in the above training set list of sources in order to use this dataset for evaluating the CADEPI classification task fairly. Table (10) shows the dataset sources and size.

Table 10: collaborative dataset sources

Sources	Economic	Political	Sport
Al-Fadjr (Algeria) Newspaper	1	1	11
Al-Ghad (Jordan) Newspaper	1	1	6
Dar Al-Hayat (London) Newspaper	1	1	1
Al-Mustaqbal (Lebanon) Newspaper	1	1	1
Al-Shames (Libya) Newspaper	1	1	1
Al-jazerah.net TV			7
Al-Messa (Algeria) Newspaper	-	-	5
Al-Ahram (Egypt) Newspaper	-	-	6
Al-Riyadh (Saudi) Newspaper	-	-	7
El-Shark (Lebanon) Newspaper	-	-	3

Emarat Alyoum (UAE) Newspaper	-	-	5
Jamahir (Syria) Newspaper	-	-	5
Taakhi (Iraq) Newspaper	-	-	4
Al-raya (Qatar) Newspaper	1	-	-
BBC Arabic	1	-	-
CNN Arabic	1	-	-
Others	6	16	-
Total	15	21	57

In this project stage, we hope to build a standard Arabic corpus accessible to all researchers who are concerned with developing machine learning tasks in the Arabic language. Essentially, this is to make the evaluation and comparison between different researcher's results as fair as possible in order to measure the improvement in this field efficiently.

4.4. Classifier models

This section shows the classification task models in two main stages. The first stage is the pre-processing of the data set, which is an early stage that deals with data set preparation by applying some features such as General Normalization, Arabic Normalization, removing stop words, light stemmers and the parts of speech technique. The second stage uses the Naïve Bayes classifier as one of the popular machine learning algorithms with and without these features to observe their influence during the classification task. Figure (7) shows these stages and the processing of the CADEPI classifier models.

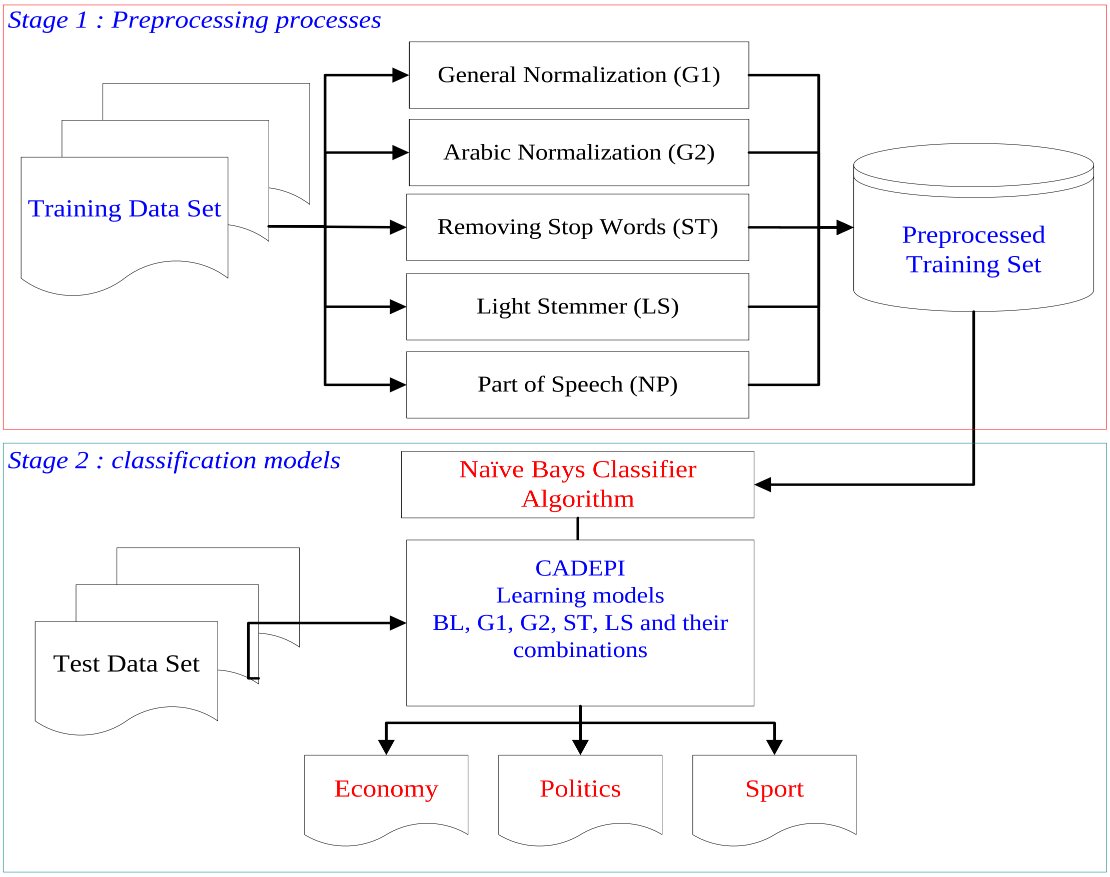

Stage 1 : Preprocessing processes
Training Data Set
General Normalization (G1)
Arabic Normalization (G2)
Removing Stop Words (ST)
Light Stemmer (LS)
Part of Speech (NP)
Preprocessed Training Set
Stage 2 : classification models
Naïve Bays Classifier Algorithm
Test Data Set
CADEPI
Learning models
BL, G1, G2, ST, LS and their combinations
Economy
Politics
Sport

Figure 7: CADEPI classifier stages and models

In the following sections, these main stages are explained further by providing their technical points of view and their functions.

4.4.1. Pre-processing of the data set and the features

In this section, five features or pre-processing techniques are described by identifying their functions and their processes during the CADEPI project classification models. Working hard to increase the quality of the pre-processing stage will undoubtedly increase the efficiency of the classification techniques.

- **General Normalization (G1)**

  The general normalization feature converts each word to the normal form which means removing any punctuation such as full stops, question marks, brackets etc… from the beginning or end of words. Table (11) shows some examples for more clarification.

Table 11: General Normalization process examples

Words before general normalization	Translation	Words after general normalization
الاقتصاد؟	Economy	الاقتصاد
التالي:	following	التالي
(مؤسسة)	organization	مؤسسة
السياسة.	Politic	السياسة
<الرياضة,	Sport	الرياضة

- **Arabic Normalization (G2)**

The Arabic Normalization feature is the method of unification of different forms of the same letter into the normal one (standard form) as follows:

1. Normalize أ, إ, آ to ا.
2. Normalize ـة to ـه.
3. Normalize ى to ي.
4. Normalize the sequence ء ي and the sequence ء ى to ئ (Syiam et al. 2006).

Table 12: Arabic Normalization process examples

Words before Arabic normalization	Translation	Words after Arabic normalization
أحمد	Ahmed	احمد
الاقتصادية	Economical	الاقتصاديه
مصطفى	Mustafa	مصطفي

- **Construction of the stopword list (ST)**

The Arabic language is rich in public words which are not connected to specific subjects or categories (stopwords). For example conjunctions, disjunctions, prepositions and pronouns (Elkourdi et al., 2004); thus creating a list containing stopwords is a useful feature to avoid unimportant words during the parsing technique. In this research, I have discovered two Arabic stopword lists. The first list contains 162 words (Savoy, 2010). The second list contains 12,201 words (Zerrouki & Amara, 2009).

However, in the CADEPI project, these lists were refined and combined to achieve a highly accurate final stopword list containing 10,747 words.

The English stopword list has 360 words; however the list of Arabic stopwords is much larger than the English stopword list for three reasons. First, the Arabic stopword list contains stopwords with all their possible forms. For example, the Arabic word for 'pronoun' has four forms:

1. هاتان (feminine, nominative).
2. هاتين (feminine, genitive/accusative).
3. هذان (masculine, nominative).
4. هذين (masculine, genitive/accusative). For more details (see section 1.3).

Second, pronouns and prepositions are sometimes connected (Chen & Gey, 2002). Third, suffixes and prefixes increase the number of Arabic words as we can see in the following section.

The average number of Arabic word forms is around 26.5 including suffixes and prefixes. It is quite large when compared with English which has 3 forms and French which has 3.5 (Ben et al., 2010). Table (13) shows some examples:

Table 13: Arabic word forms examples

Forms	1	2	3	4	5	6	7 ….
Arabic word	السياسة	سياسة	والسياسة	فالسياسة	كالسياسة	السياسيات	السياسيين
Translation	The Politics	A Politics	And Politics	Then Politics	Such as Politics	Politician Feminine	Politician Masculine

- **Light stemmer(LS)**

The stemming technique is the process of word root extraction which enables a return to the word in its original form (roots). Reducing the data set and applying classifier algorithms to significant data only, will enhance the performance time and the classification accuracy. There are two types of stemming process: strong stemming (i.e. word root extraction) and light stemming (i.e. removing some defined prefixes and suffixes from the word only). Using light stemming is more effective during the classification task than strong stemming for the following reasons. The main reason for using light stemming is that several word variants do not have similar meanings or semantics. Nevertheless, these word variants are generated from the same root. Accordingly, root extraction algorithms affect the meanings of words (Al-refai et al., 2007).

In the CADEPI project a light stemming process is applied as follows:

- A set of prefixes (بال, كال, فال, ال, لل, و) is removed. Afterwards, word length is checked if it is less than three letters. This prefix is considered a main part of the word hence the removed prefix is returned to the word.

- A set of suffixes (ـه,هما, كما, ات, يه, ته, تي, ان, ون, ين, هم, هن, ها, نا, وا ,كم, كن , ي) is recursively removed from the end of the word. The recursive removal process removes the longest suffix first, then the shorter, since suffixes are mainly a compound of pronouns, gender and number suffixes. For instance the words (مكتباتهم) (Their libraries) has a composite suffix (اتهم) which is made up of two parts (ات) for feminine plural and the pronoun (هم). Furthermore, as in the previous step, word length is checked if it is greater than three letters with the purpose of not removing the main part of the word (Syiam et al., 2006).

- **Parts of speech (NP)**

Categorization of words into parts of speech is a useful task that will simplify the word domain and provide important information needed to classify the document. In this case, nouns and proper nouns are considered as providing highly important information for classification. The El-halees (2007) study achieved significant progress in classification accuracy by implementing classifier algorithms to nouns and proper nouns only (El-halees, 2007). Consequently, parts of speech (POS) were studied by focusing on two types: the POS Stanford POS Tagger, v. 3.0 - 2010-05-10 and the transformation-Based Learning POS tagger.

The Stanford POS Tagger uses a Maximum Entropy parts of speech tagger and the best resultant accuracy for this tagger on the Penn Treebank is 96.86% overall (Toutanova & Manning, 2000). The Transformation-based Learning parts of speech tagger (TBL tagger) was developed for the Arabic language and achieved an accuracy of 96.9% (AlGahtani et al., 2007). As result the TBL post tagger was used in the CADEPI project to include nouns and proper nouns only and exclude the others parts of speech during the Arabic contents classification task.

4.4.2. The Naïve Bayes classifier algorithm

The Naïve Bayes classifier algorithm is investigated in this project because it is not sensitive to either word position or text direction. As a consequence it is helpful as long as the Arabic language's direction is from right to left. Figure (8) shows the NB algorithm process. For more details see section 3.2.3 in the previous chapter.

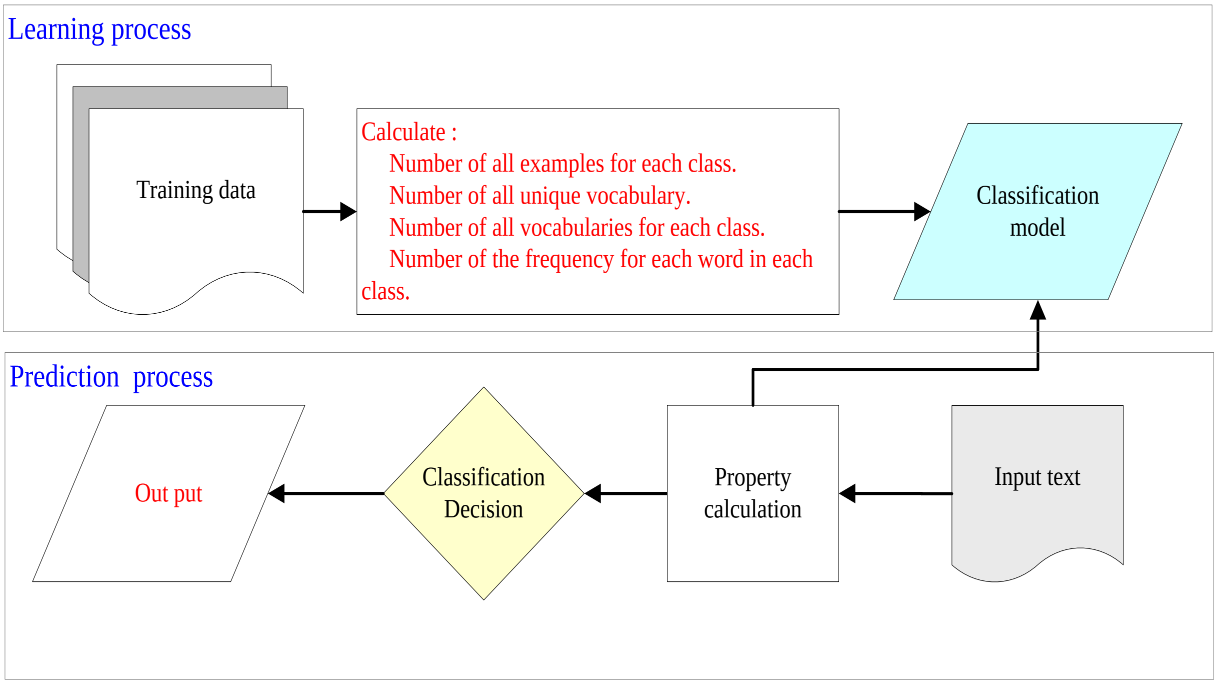

Learning process
Training data
Calculate :
Number of all examples for each class.
Number of all unique vocabulary.
Number of all vocabularies for each class.
Number of the frequency for each word in each class.
Classification model
Prediction process
Out put
Classification Decision
Property calculation
Input text

Figure 8: Naïve Bayes text classifier algorithm procedure

4.4.3. Challenges

Several challenges have appeared during the data collection and implementation stages (e.g. collecting a substantial number of Arabic documents while considering the significance of its quality). Furthermore, these documents needed refining several times in order to make sure their contents were strongly related to the project classes. Choosing the appropriate tools for the pre-processing stage (e.g. light stemming tool, parts of speech tool) is a challenge because they needed further study and efficient testing before applying them in the CADEPI project and measuring their performance on the dataset.

4.5. Experimental methods for the classification task

In the classification task of the CADEPI project, three experimental methods were used to evaluate their learning techniques. The first method was the 10-fold cross-validation method, which is the standard machine learning method for evaluating the performance of classification algorithms. The second method was the reverse 10-fold cross-validation method which was created during the project to assess the performance of classification algorithms with a limited dataset. The third method was the normal test, which uses the collaborative dataset to evaluate the CADEPI classifier efficiency in the real world dataset.

Essentially, these methods were used to evaluate (NB) as a baseline with pre-processing features in different models of classifications. Each method is described in the sections which follow.

4.5.1. The 10-fold cross-validation method

In the 10-fold cross-validation method the dataset is first partitioned into 10 equally (or almost equal) sized partitions or folds. Ten iterations of training and validation are performed in each iteration. A different partition of the data is held out for validation while the remaining 9 partitions are used for learning. Figure (9) demonstrates an example where the darker sections of the data are used for training and the lighter sections for validation (Refaeilzadeh et al., 2008).

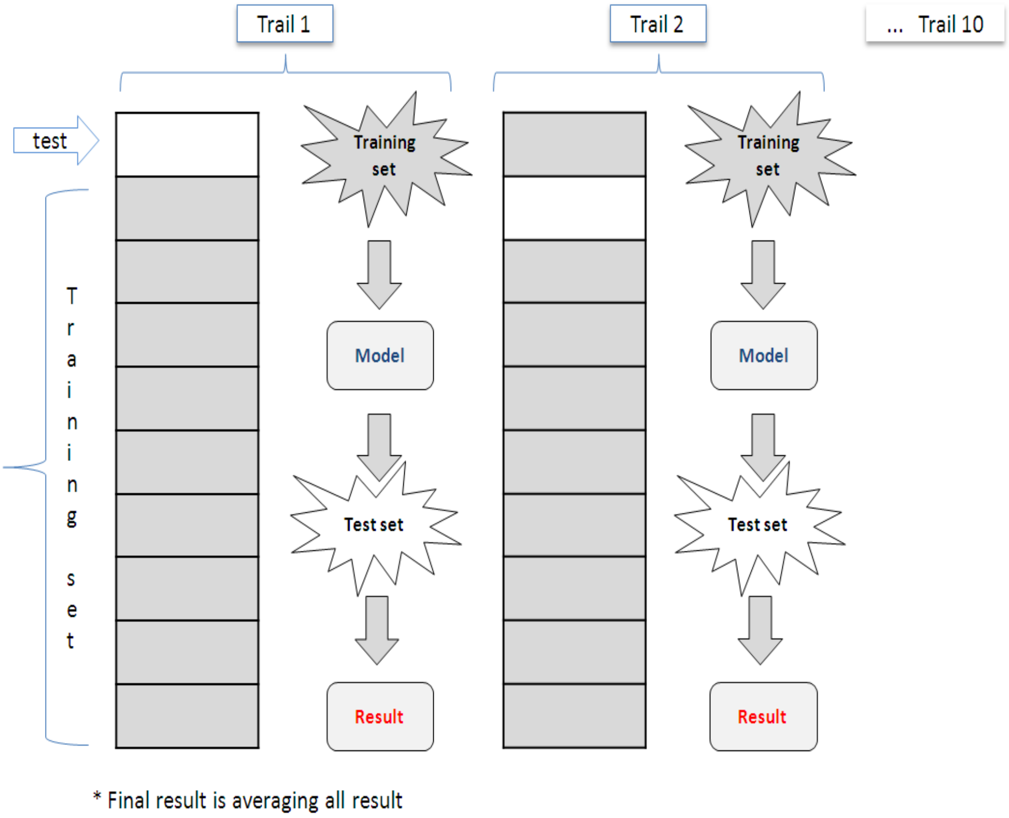


Figure 9: 10-fold cross-validation

For that reason, particular codes were implemented in order to divide the CADEPI dataset into 10 partitions randomly with 90% for training and 10% for validation (test data). Then the f-measure is calculated for each iteration in order to complete the final step which is to calculate the average of the f-measure values for each CADEPI classification model that is available for the evaluation and comparison stage.

These data are used in the t-test as a statistical confidence test to evaluate the difference in performance between each of the classification models.

4.5.2. The Reverse 10-fold cross-validation method

The reverse 10-fold cross-validation method is a new experimental method which was created during the CADEPI project. It is similar to a standard 10-fold cross-validation technique; however, the difference is in the amount of training and validation data which transfers to 10% as a training set and 90% as a validation set. It performs to evaluate the classification algorithm (NB) in limited example of training set and large example of validation data that to show the performance of pre-processing features on the learning algorithm (NB) obviously. Figure (10) shows an example where the grey sections of the data are used for training (10%) whilst the non-shaded sections are used for validation (90%).

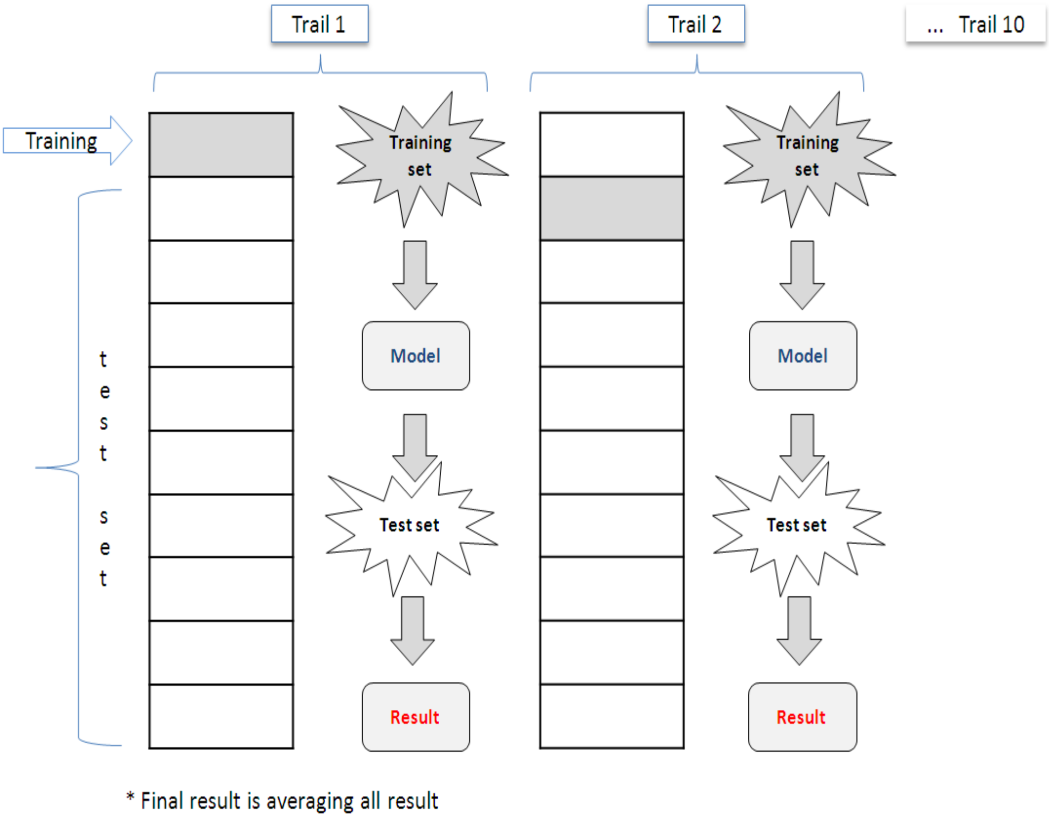


Figure 10: Reverse 10-fold cross-validation

Furthermore, particular codes were created in order to apply this new method by dividing the CADEPI dataset into 10 partitions randomly with 10% for training and 90% for validation (test data). The f-measure value per iteration is then calculated; this leads to the final step of calculating the average of the f-measure values for each CADEPI classification model that is available at the evaluation and comparison stage. These data are used by the t-test to evaluate the statistical significance of the difference in performance between each classification model when this new method is used.

4.5.3. The Normal test

In this experimental method, the data set, which is collaboratively gathered as an unbiased data set in Table 9, is used in an attempt to measure the CADEPI classifier with its ideal classifier model performance in the real world application.

4.6. Performance measures for classification task scenarios

For this study recall, precision and F-measure statistical methods are used as standard evaluation measures for classification tasks. Also, the t-test statistical test is used to compare performances between CADEPI project models. (For a more detailed description, see section 3.6).

In a classification task scenario, recall is the number of correctly classified documents in the relevant class divided by the total number of existing documents in the relevant class. Precision refers to the number of correctly classified documents by classification in the relevant class divided by the total number of classified documents in relevant class. The F-measure is a measure of a test's accuracy. It considers both the precision and the recall of the test (Mccallum & Nigam, 1998).

```
For 13 CADEPI models :
    For three classification categories
        1. Recall = Number of correct classified documents / total number of existing documents.
        2. Precision = Number of correctly classified documents / total number of all classified documents.
        3. F-measure = 2 *(Precision * Recall)/ (Precision +Recall).
    End.
End.
```

In this project, the comparison technique is used in a simple two-tailed, paired sample t-test at significance level p = 0.05. The aim is to determine whether the difference between CADEPI's classifier model F-measure score is statistically significant for selecting the ideal model to use as a default classifier model in CADEPI. There are different tools for the t-test calculation; however in this case, the t-test function in Microsoft Excel is used to calculate the p value.

4.7. Name extraction models

This section describes the characteristics of Arabic name extraction in the CADEPI project using three models. First, the Arabic name dictionary (ANDic) only is applied in the name extraction task in order to evaluate its efficiency since it is the first time it has been used (Dic). Second, the ANDic dictionary with its parts of speech technique, which is the pre-processing feature, is applied in order to filter the Arabic contents into nouns and proper nouns (Dic&NP). Third, the ANDic dictionary with its parts of speech technique is applied to filter the Arabic contents into nouns, proper nouns and adjective (Dic&NPJ). Figure (11) shows these stages and the CADEPI name extraction processing model.

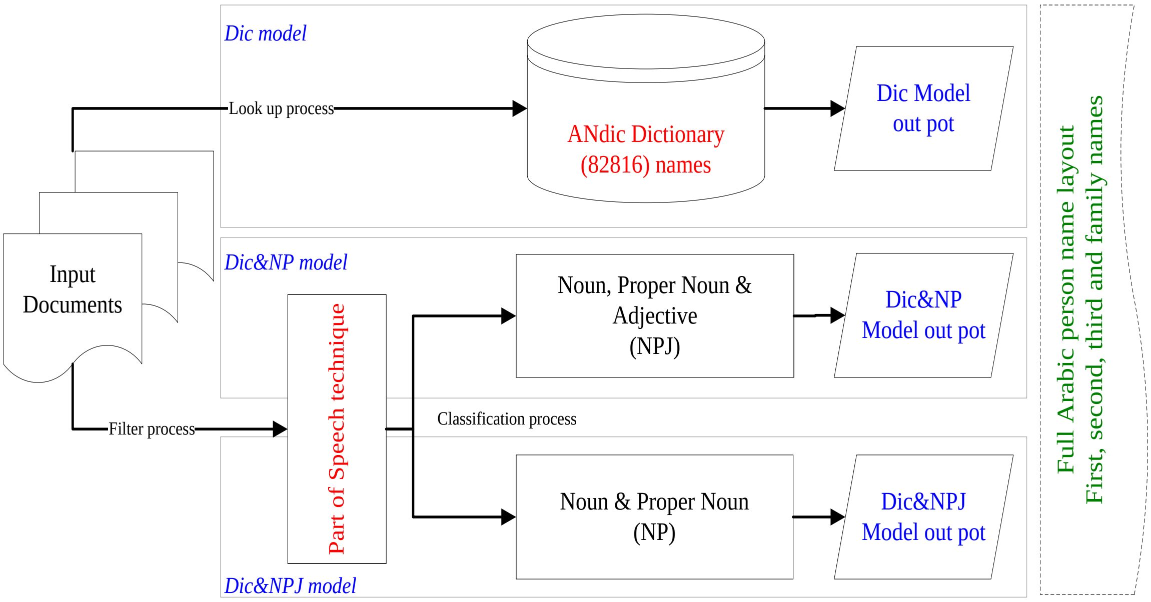
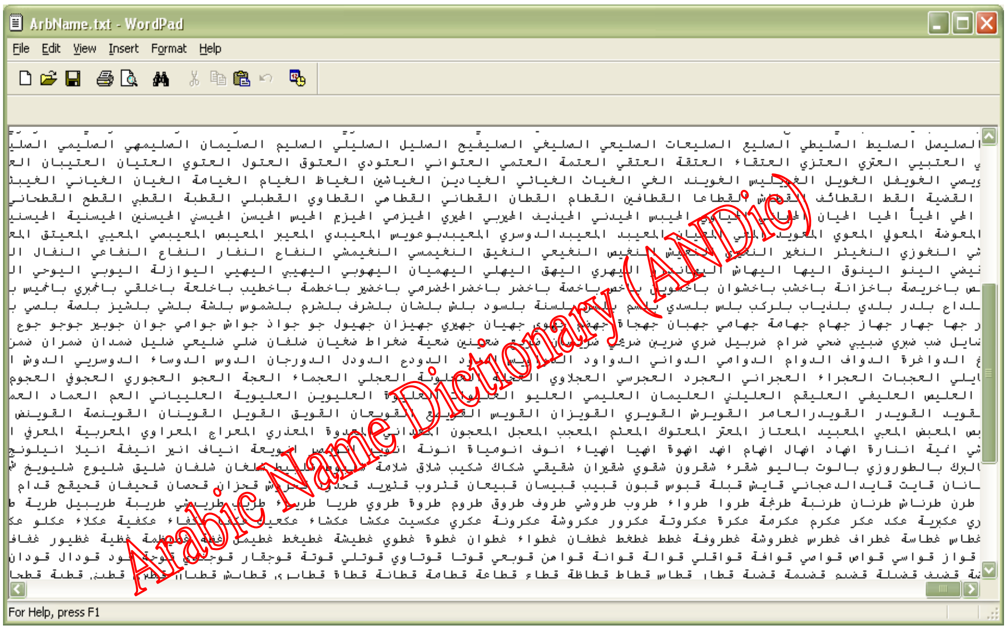


Figure 11: CADEPI Name extraction models

4.7.1. Dictionary

At an early stage in the preparation of this project, a database of Arabic personal names was collected from different sources of government databases for treating the ambiguity between proper nouns and personal names; this will contribute to an increase in the personal name extraction task efficiency. This database contains approximately 304,638 Arabic names. It is a large number for personal names; the reason behind this is that it contains all possible name forms as we have seen in sections 2.4 and 3.4.1 above so one name has approximately 3 or 4 different forms.

During these project stages, the (ANDic) dictionary is refined by removing all compound name forms that have united forms. For example Abduallah which is a very common Arabic name, has two forms as compound forms, "عبد الله" which is the space between the first part Abdu and the second part Allah, and a united form "عبدالله" where there is no space between the first and second part. These formats are widespread in Arabic names such as Abdu Alrahman, Abdu Alaziz, Abdul baset , Esam Aldeen etc…thus, keeping the first part itself as a name in the ANDic dictionary and removing other parts as long as the second part exists in the dictionary as a name itself.In our example Abdu will exist and Allah will be removed because Allah is included in the dictionary as a name itself.   This process, for all same name styles, reduces the number of names in the dictionary substantially number; hence, the content of the ANDic dictionary was reduced to 82816 Arabic names. Evaluating its efficiency during the CADEPI project became an essential task to facilitate development and make it more suitable in the real world. Figure (12) shows some samples of the Arabic name dictionary (ANDic).


ArbName.txt - WordPad
File Edit View Insert Format Help
Arabic Name Dictionary (ANDic)

الصليصل الصليط الصليطي الصليع الصليعات الصليعي الصليغي الصليفيح الصليل الصليلي الصليم الصليمان الصليمهي الصليمي الصلي
ي العتيبي العتري العتزي العتقاء العتقة العتقي العتمة العتمي العتواني العتودي العتوق العتول العتوي العتيان العتيبان الع
ويمي الغويفل الغويل الغويليس الغويند الغي الغياث الغياثي الغيادين الغياشين الغياظ الغيام الغيامة الغيان الغياني الغيبة
القضية القط القطائف القطاش القطاعا القطافين القطام القطان القطاني القطامي القطاوي القطبلي القطبة القطبي القطح القطحاني
الحي الحيأ الحيا الحيان الحياني الحياوي الحيبس الحيدني الحيذيف الحيربي الحيري الحيزمي الحيزم الحيس الحيسن الحيسني الحيسنين الحيسنية الحيسني
المعوضة المعولي المعوي المعويد المعي المعيان المعيبد المعيبدالدوسري المعيبدبوعويس المعيبدي المعيبر المعيبص المعيبصي المعيبي المعيتق المع
شي النغوزي النغيثر النغير النغيري النغيش النغيص النغيعي النغيق النغيمسي النغيمشي النفاع النفار النفاع النفاعي النفال ال
نيضي الينو الينوق اليها اليهاش [illegible] اليهري اليهق اليهلي اليهميان اليهوبي اليهيبي اليهيي اليوازلة اليوبي اليوحي ال
ص باخريصة باخزانة باخشب باخشوان باخشوين باخص باخصة باخضر باخضرالحضرمي باخضير باخطمة باخطيب باخلعة باخلقي باخميري باخميس ب
لداع بلدر بلدي بلذياب بلركب بلس بلسدي بلسم بلسن بلسنة بلسود بلش بلشان بلشرف بلشرم بلشموس بلشة بلشي بلشيز بلصة بلصي ب
ر جها جهار جهاز جهام جهامة جهامي جهبان جهجاة جهجو جهوي جهيان جهيري جهيزان جهيول جو جواذ جواش جوامي جوان جوبير جوجو جوع
ضايل ضب ضبري ضبيبي ضحي ضرام ضربيل ضري ضرين ضريحي ضريسان ضرير ضعنين ضعية ضغراط ضغيان ضلفان ضلي ضليعي ضليل ضمدان ضمران ضمن
غ الدواغرة الدواف الدوام الدوامي الدواني الدواود الدواريس الدوايد الدودج الدودل الدورجان الدوس الدوساء الدوسريي الدوش ا
ايلي العجبات العجراء العجراني العجرد العجرسي العجلاوي العجلان العجلونة العجلي العجماء العجة العجو العجوري العجوفي العجوم
العليص العليفي العليقم العليلي العليمان العليمي العليو العليوات العليوة العليوين العليوية العلييياني العم العماد العم
قويد القويدر القويدرالعامر القويرش القويري القويزان القويس القويع القويعان القويق القويل القوينان القوينمة القوينض
بص المعبض المعي المعبيد المعتاز المعتر المعتوك المعنم المعجب المعجل المعجون المعداني المعدوة المعذري المعراج المعراوي المعربية المعرق ا
شي انمية انارة انهاد انهال انهام انهد انهوة انهيا انهياء انوف انومياة انونة انوفي انوصر انويعة انياف انيفة انير انيلا انيلونج
البرك بالطوروزي بالوت باليو شقرء شقرون شقوي شقيران شقيقي شكاك شكيب شلاق شلامة شلبوط شلبيط شلغان شلفان شليق شليوع شليويخ ش
انان قايت قايدالدعجاني قايش قبلة قبوس قبون قبيب قبيسان قبيعان قنروب قنيريد قحدان قحروش قحزان قحصان قحيفان قحيقح قدام
طرن طرناش طرنان طرنبة طرفجة طروا طرواء طروب طروشي طروف طروق طروم طروة طروي طريا طريباء طريب طريبشي طريبة طريبيل طرية ط
ري عكبرية عكد عكر عكرم عكرمة عكرة عكروتة عكرور عكروشة عكرونة عكري عكسيت عكشا عكشاء عكعيك عكفاء عكفية عكلاء عكلو عك
غطاس غطاسة غطراف غطرس غطروشة غطروفة غطط غطغط غطفان غطواء غطوان غطوة غطوي غطيشة غطيغط غطيمل غظة غظمة غظية غظيور غفاف
قواز قواسي قواص قواصي قوافة قواقلي قوالة قوانة قواهن قوبعي قوتا قوتاوي قوتلي قوتة قوجقار قوجني قوجة قود قودال قودان
قضيف قضيلة قضيم قضيمة قضية قطار قطاس قطاط قطاظة قطاء قطاعة قطامة قطانة قطاة قطاير قطايش قطبان قطرم قطني قطية قطحا

For Help, press F1

Figure 12: Arabic names dictionary (ANDic)

4.7.2. Parts of speech

This technique, which refers to the classification of words into parts of speech (noun, proper noun, adjective and verb), is a useful method to reduce the word domain and include only important information - in this case nouns, proper nouns and adjectives only. Therefore, applying this technique is extremely important because it is also used in the first part of the CADEPI project which deals with Arabic document classifications, however in this section the adjective which is one of the Arabic word's part of speech is added because large numbers of Arabic names are adjectives.

4.7.3. Challenges

Several challenges have appeared especially during the preparation and the integration stage of the data in the Arabic name dictionary ANDic because it needed refining and testing several times before the ideal presentation could be chosen (data base table or file) during the name extraction task. Moreover, increasing its performance in order to be highly efficient and meet the CADEPI project needs was a real challenge.

Also the language direction of the Arabic language is from right to left. It was a challenge because all programming languages read file contents from left to right. Practically, it was a problem during the gathering of full Arabic name components in this style (first, second, third … family names) because the last name will read first, and so on until the first name. Detecting the last and first name was a problem; however it was solved partially: any word which is not a name and which is not contained in the dictionary becomes an end word break and each name gathered before it is a full name. Afterwards one starts again to find other full names in the contents.

4.8. Experimental methods for name extraction tasks

In the CADEPI project, 10 files were chosen randomly out of the training set as assessment measurement files. They were chosen to extract all Arabic full personal names from their contents manually. Next, the CADEPI name extraction models, the Dic, the Dic&NP and the Dic&NPJ, see Figure (11), were applied to extract the same 10 files automatically. After that, the accuracy of these models was calculated using performance measures which are explained in the following sections in order to compare and evaluate their performance.

4.9. Performance measures for name extraction task scenarios

For this study, recall, precision and F-measure statistical methods were used as standard evaluation measures for NLP applications. These methods were described above in sections 3.6.1.

For these 10 files, which were randomly chosen, all full Arabic names and their contents were determined as measurement values. Consequently, three name extraction models were applied and their results were compared using measurement values to calculate recall, precision and f-measures for each model as follows:

For three models:

1. *Recall = Number of correct full names extracted / total number of existing full names.*
2. *Precision = Number of correct full name extracted / total number of all extracted full name.*
3. *F-measure = 2 *(Precision * Recall)/ (Precision +Recall).*

End.

4.10. Summary

This chapter described the methods used in designing this project, it involved its objectives, the classification of Arabic contents and the extraction of Arabic full personal names. The data needed, pre-processing features and learning Algorithms and how they have been applied during the project's stages were explained. The CADEPI project's models of the classification task and the name extraction task were illustrated. The performance measures and how the project's models were assessed were described.

All of these techniques have been investigated to realize the most suitable models for Arabic content classification and name extraction. The next chapter will illustrate the experimental results of the CADEPI project.

5. Chapter 5: Experimental Results

5.1. Introduction

In this chapter, the Arabic dataset described above has been used for training and testing the CADEPI project objectives. Different methods such as the 10-fold cross-validation and the reverse 10-fold cross-validation method were used in order to evaluate the Arabic text classifier performance using different models (13 CADEPI models) and choosing the ideal CADEPI model which it tested in the real world as an extra evaluation method. Name extraction experimental results, (another aim of this project with the new dictionary), were evaluated using 3 CADEPI models in a separate section. Furthermore, statistical T-test results are presented in order to contrast the CADEPI models and assess the difference between their performances.

5.2. Classification of the content of Arabic Document experiments

As we mentioned earlier, classification of the contents of Arabic documents is tested using different models. These models involved some pre-processing steps as external features which were working with the baseline algorithm (NB). These features, which are mentioned above, are:

- Naïve Bayes classifier (NB) as baseline model (BL).
- NB with General Normalization as model (G1).
- NB with Arabic Normalization as model (G2).
- NB with Removing stop words as model (ST).
- NB with Light stemmer as model (LS).
- NB with parts of speech technique as model (NP)
- NB with General Normalization and parts of speech technique as model (G1,NP)
- NB with General Normalization and Removing stop words technique as model (G1,ST).
- NB with General Normalization and Light stemmer technique as model (G1,LS).
- NB with General Normalization, Removing stop words and Parts of speech technique as model (G1,ST,NP)

- NB with General Normalization, Light stemmer and Parts of speech technique as model (G1,LS,NP)
- NB with Removing stop words and Parts of speech technique as model (ST,NP).
- NB with all stand alone features together as model (All).

Thirteen models were investigated using two methods and the ideal model, which achieved a higher f-measure value and a lower significant p value during the evaluation, was investigated using a further method. The results are described using three methods as follows:

5.2.1. The 10 cross validation experiment results

For comparison, the 10-fold cross validation method was applied to 13 different models including the baseline algorithm NB as well as the NB with a different combination of pre-processing features. The F-measure value was calculated for each of the classification classes and for the model overall. Moreover, the t-test was applied in order to statistically compare the performance difference between the baseline model and other models.

Table 14: 10 cross validation experiment results of the F-measure.

	Features	Economy	Politics	Sport	overall
Singular	BL : Base Line	98.44	98.68	98.47	98.53
	G1 : General Normalization	98.53	98.64	98.69	98.62
	G2 : Arabic Normalization	98.21	98.27	98.64	98.37
	ST : Stop Words	98.29	98.08	98.74	98.37
	LS : light stemmer	98.71	98.31	98.97	98.66
	NP : Noun and proper Noun	99.12	98.35	99.06	98.84
Components	G1,NP	99.15	98.76	99.22	99.04
	G1,ST	98.64	98.81	99.08	98.84
	G1,LS	99.31	98.86	99.29	99.15
	G1,ST,NP	99.47	99.27	99.26	99.33
	G1,LS,NP	99.30	99.10	99.41	99.27
	ST,NP	98.81	98.78	99.33	98.98
	All	98.68	98.45	98.13	98.42

Table (14) shows that the parts of speech technique which filters the contents and chooses the noun and proper noun words (NP) outperforms every other single feature by a 98.84 f-measure value. The second best is the light stemmer technique (LS), which removes the common prefix and suffix of Arabic words by a 98.66 f-measure value.

However, the statistical results of these conclusions (t-test values) were not significant during comparisons of the performance differences between the baseline model and the baseline and involved some singular features. Table (15) shows the t-test result and also that the (BL) compared with (NP) feature was 0.153376 p value which is the lowest value.

Table 15: The 10 fold cross validation experiment result of the T-test value for singular features

T test 1	P value
BL with G1	0.715138
BL with ST	0.506507
BL with NP	0.153376
BL with LS	0.509553

Furthermore, any combination of features, which includes the NP, indicates substantial performance. For example, the combination of General Normalization, removing stopwords and choosing nouns and proper noun words using the parts of speech technique (G1, ST, NP) outperforms all other models with a 99.33 f-measure value. In addition the combination of General Normalization, light stemmer and choosing noun and proper noun techniques (G1, LS, NP) ranks second with a 99.27 f-measure value. It is very close to the first model as we can see in Figure (13).

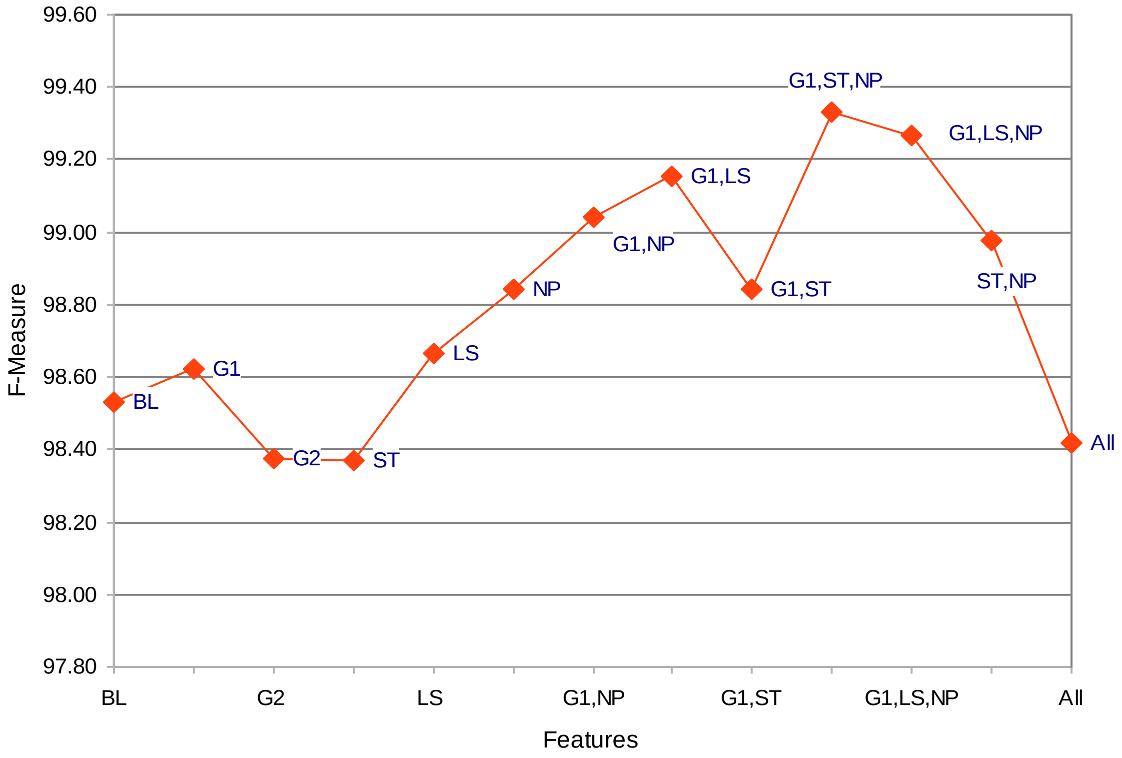


Figure 13:10 Cross validation experimental results

To test these conclusions statistically, a t-test was applied in order to compare the performance differences between the baseline model and some combinations of features. Consequently, a combination of (G1,ST,NP) and (G1,LS,NP) compared with the baseline BL showed significant p value results; they were 0.000377 and 0.000852 respectively. Table (16) shows the p value for some model comparisons.

Table 16 : 10 fold cross validation experimental result of the T-test value for combination features

T test 1	P value 1
BL with G1STNP	0.000377
BL with G1LSNP	0.000852
BL with ALL LS	0.638152
G1,ST,NP with G1LSNP	0.723947

Generally, these results show that the parts of speech technique as a pre-processing feature has played a major role in classifying Arabic document contents – the primary CADEPI project goal – as a single feature model or with models of combination features. It reduces the input of less important words during the classification process and thus it helps the NB classifier focus on the significant parts of speech which are noun and proper noun words. Consequently, that led to an increased performance in the classification task in the CADEPI project. Arabic normalization (G2), which is a pre-processing feature to convert some Arabic letter forms into normal forms, did not reflect a substantial difference in performance during this project's experiments because it affects some word forms strongly which led to a decreased performance in classification tasks as we can see in Figure (13).

To summarize this stage of the experiments, the 10-fold cross validation method presented (G1,ST,NP) as the ideal model so far since other different experimental methods are investigated in the following sections.

5.2.2. Reverse 10 cross validation experiments result

In the CADEPI project, this new experimental method with the machine learning algorithm was created to investigate the classification task with a limited number of data sets. Section (4.5.2) describes this method's aims and procedures particularly. This new method was applied to 13 different models including the baseline NB algorithm and NB with different combinations of pre-processing. The F-measure value was calculated for each classification class and for the models overall. The t-test was applied as well in order to compare statistically the performance difference between the baseline model (BL) and other models.

Table (17) shows that the light stemmer technique which removes the common prefix and suffix of Arabic words (LS) outperforms every other single feature by 97.26 using the F-measure with NB. The second best is the part of speech technique which filters the contents and chooses noun and proper noun words (NP) by 96.40 F-measure value.

Table 17: Reverse 10 cross validation experiment results.

	Features	Economy	Politics	Sport	over all
Singular	BL	93.51	91.97	95.31	93.59
	G1	95.27	94.25	96.03	95.18
	G2	93.99	92.67	95.73	94.13
	ST	96.00	94.80	97.79	96.20
	LS	97.18	96.21	98.39	97.26
	NP	96.29	95.19	97.71	96.40
Components	G1,NP	97.08	95.85	97.76	96.90
	G1,ST	96.49	95.49	97.97	96.65
	G1,LS	97.47	96.63	98.50	97.53
	G1,ST,NP	97.43	96.40	98.14	97.32
	G1,LS,NP	98.01	96.88	98.25	97.71
	ST,NP	97.04	96.11	98.07	97.07
	All	97.51	96.17	97.60	97.09

These results were tested statistically using the t-test measurement method and compared the performance difference between the baseline model (BL) and some singular features in this new experimental method. Essentially, their t-test values were significant for all singular features contrasting with the baseline algorithm. For example, comparing the light stemmer technique (LS) with the baseline algorithm (BL) which was 3.77E-11 ≈ 0.0 p value and (NP) with (BL) was 8.76E-16 ≈ 0.0 p value as we can see in Table (18). Therefore, these features significantly progressed the performance of the Arabic contents classification task in the CADEPI project.

Table 18: Reverse 10 fold cross validation experimental results of the T-test value for singular features

T-test	p value
B1 with G1	8.44E-06
BL with ST	3.77E-11
BL with NP	8.76E-16
BL with LS	2.50E-21

Furthermore, a combination of features was investigated; for example, the combination of General Normalization, light stemmer techniques and choosing noun and proper noun words using the parts of speech technique (G1, LS, NP) outperformed all other models by 97.71 f-measure value. The second best combination was the General Normalization and light stemmer (G1,LS) by 97.53 f-measure value and the combination of General Normalization, light stemmer, removing stopwords and choosing noun and proper noun words by part of speech technique (G1, ST, NP) was third.(by 97.32 f-measure value). They are very close together as we can see in Figure (14).

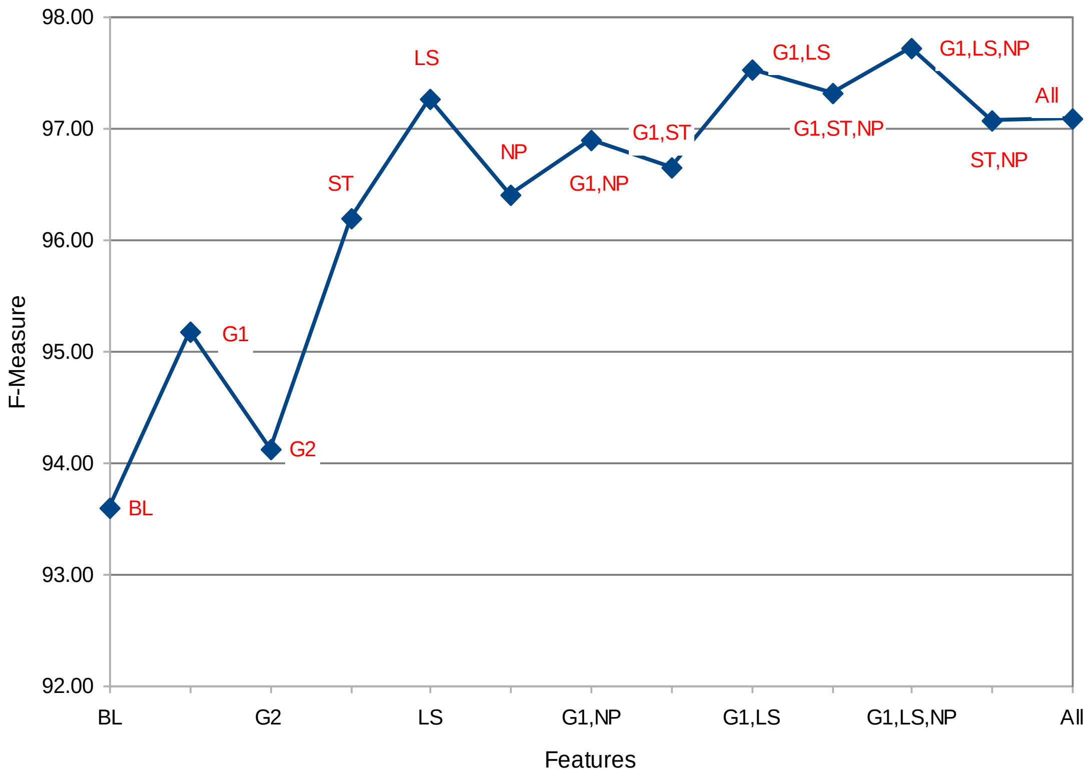


Figure 14: Reverse 10 cross validation experimental results

To test these results statistically, the t-test was applied in order to compare the performance difference between the baseline model and these combinations of features. Therefore, combinations of (G1,LS,NP) and (G1,ST,NP) compared with the baseline (BL) were significant; they were 6.30E-26 and 7.05E-23 respectively. Table (19) shows the p value for some models and comparative result.

Table 19: Reverse 10 fold cross validation experimental results of the T-test value for combination features

T-test	p value
B1 with G1STNP	7.05E-23
BL with G1LSNP	6.30E-26
BL with ALL	2.13E-21
G1STNP with G1LSNP	6.36E-03

In this new experimental method (10-fold cross validation), the light stemmer technique as a pre-processing feature has played a significant role in classifying Arabic document content tasks because it reduces the input (words domain) from repeated word forms which have the same meaning during the classification process. Therefore, the classifier algorithm (NB) focuses on the main word form that has a full word meaning which is very important during any text classification task. Thus, this technique led to increasing the performance of the classification task notably in this CADEPI project. Moreover, all pre-processing features reflect this substantial performance in this project, compared with (BL) in this new experimental method, as we can see in Figure (14).

To summarize, this stage of the experiments using the new method (reverse 10-fold cross validation) presented (G1,LS,NP) as an additional ideal model so far since there is one further experimental method discussed in the following sections.

5.2.3. Normal test method results

The previous two different experimental methods proposed two different models of combined features as ideal models (G1, ST, NP) and (G1, LS, NP) in the Arabic contents classification task. More investigative methods to test these models in real world applications are needed by using collaborative data sets in order to determine which model is the ideal classifier model of the CADEPI project. This collaborative data set was gathered from different sources as discussed in section (4.3).

Table 20: collaborative dataset (unbiased dataset) classification result

Category	Document number	G1,ST,NP		G1,LS,NP	
		No	Accuracy	No	Accuracy
Economy	14	0	100%	0	100%
Politics	21	1	95.24%	1	95.24%
Sport	57	2	96.5%	2	96.5%
Overall	92	3	96.7%	3	96.7%

Table (20) shows these experimental results by presenting, for each category document number, the number of misclassified files and the classification accuracy for these two classification models of the CADEPI. They achieved the same performance in each classification category by 96.7% overall accuracy and the same number of the misclassified files which is 3 files out of 92 files. One file in the politics class was classified as Economy and two files classified as politics were classified in the sport class.

The verification process was completed with all three files in order to know the reasons behind this misclassification. In the politics class, the file, which was classified under economy, was about the history of politics during the two World Wars as subject; however, some words such as “analysis” and “institutes” occurred many times in the file contents where they important words in the economy class. First misclassified files in sport, which were classified as politics, are about urgent meetings of the gymnastics sports union. During this meeting some recommendations and instructions were realised so some words such as “meeting”, “members” and “union” occurred several times as they are very common words in the politics class. The second file is about general youth camps involved in some sport and social activities. They are supervised by the Jordanian military force so that words such as “militaries” and “police” are mentioned many times and they include the file contents so these words are strongly related to the politics’ class. These reasons were reasonable partly because the misclassified files are difficult for people to recognise because they have many subject intersections. However, they were noted in the algorithms’ limitations. This is an area for future study.

In general, (G1, ST, NP) and (G1, ST, NP) are ideal models in the CADEPI classifier task because they exhibit substantial performances in all the different experiments; however, it might be said that (G1, ST, NP) is an ideal model if there were enough examples in the training set which was approved during the first experimental method (10-fold cross validation). The (G1, LS, NP) model is more convenient if there is a limitation on training set examples approved during the second experimental method (reverse 10-fold cross validation). In the last experimental method – normal test – these models’ performances were equivalent.

Table 21: Comparing (G1,ST,NP) and (G1,LS,NP) performance for experimental methods results

Experimental method	**G1,ST,NP**	**G1,LS,NP**
10-fold cross validation	99.33	99.27
Reveres 10-fold cross validation	97.32	97.71
Normal test	96.7	96.7
Average	97.78	97.9

However, calculating the performance average of all experimental methods for these models showed that the (G1, LS, NP) model is the ideal model for the CADEPI project in the classification task by 97.9 as average of all experimental method performance, as we can see in the Table (21) and Figure(15) . Effectively, that referred to the part of speech technique, which filters the Arabic text into nouns and proper nouns (NP), and light stemming technique, which returns a word to its root by removing very common prefix and suffixes, working together include this ideal model as more efficient pre-processing features.

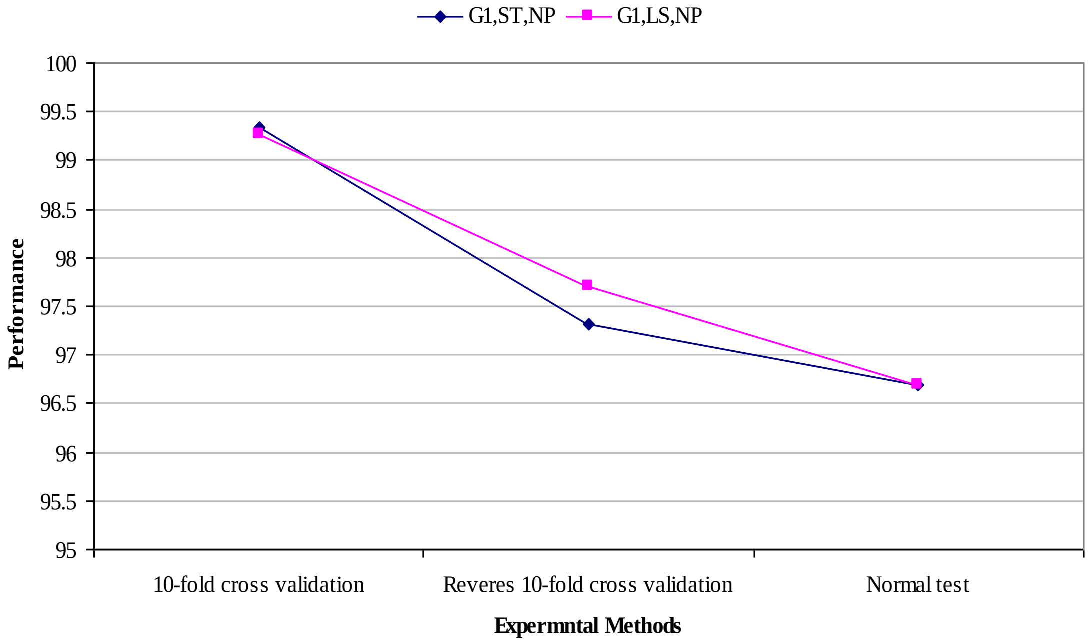


Figure 15: Comparing (G1,ST,NP) and (G1,LS,NP) models performance

5.3. Arabic name extraction experiments

In this project, extracting the Arabic personal names in the Arabic text contents was investigated using three models. The first model applyied a new dictionary (ANDic) by itself (Dic). It is used here for the first time in this project which contains 82816 Arabic names. The second model was the ANDic with the parts of speech technique as a pre-processing step to determine the nouns and proper nouns (Dic & NP). The third model applied ANDic using the parts of speech technique as a pre-processing step to determine the nouns, proper nouns and adjectives (Dic & NPJ).

Table 22: Arabic name extraction model results

name recognition	Recall	Precision	F-measure
Dic	82.45	81.10	81.77
Dic & NP	74.05	83.85	78.65
Dic & NPJ	81.49	80.53	81.01

Table (22) shows that the first model (Dic) outperforms other models by 81.77 f-measure accuracy because the Arabic name forms have different parts of speech forms as Arabic names are verbs or adjectives. However the majority of Arabic name forms are nouns. That was clear in the second model where the contents are filtered (nouns and proper nouns) by using the parts of speech technique. The accuracy value is 78.65 less than the first model. This used a dictionary only because Arabic personal names involved the adjective as a part of speech. It was approved by the third model because it involved the adjective with nouns and proper nouns and it achieved an accuracy value of 81.01. This was better than the second model. Figure (16) shows all models' performances.

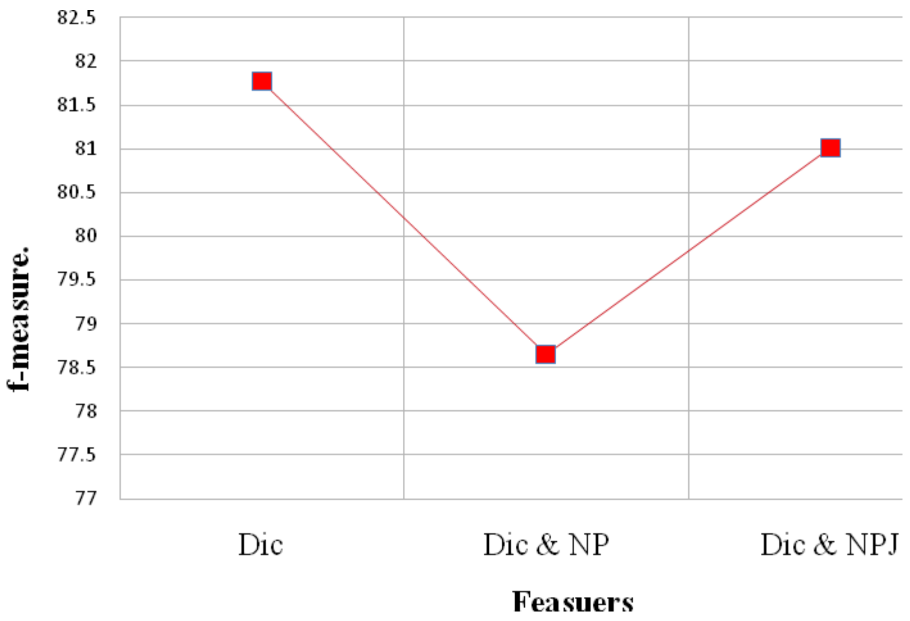


Figure 16: Arabic name extraction results

To summarize this result, the parts of speech technique's performance with Arabic name extraction is not helpful because the Arabic name is highly distributed between all parts of speech (Noun, Proper noun, adjective, verb and adverb). Thus it does not provide any progress in Arabic name extraction tasks any more than using the ANDic dictionary does by itself. Name extraction tasks in the Arabic language are very complicated tasks because of the Arabic name structure; therefore it needs more investigation to apply Arabic name rules. Although, (ANDic), as an Arabic name dictionary is used for the first time, it provides a considerable performance accuracy of 81.77 on the f-measure.

5.4. Dissection

This section discusses the project results compared with previous work that involves techniques and methods which have been used during the project. It is divided into two subsections according to the project's objectives.

5.4.1. Classification of the content of Arabic Document results

In general, the CADEPI classification process proved that the pre-processing features play a substantial part in text classification because they constrict the text contents domain into the important information in order to support the learning algorithm and ignores the less important information (noise data). They minimise the running time of the classification process and the required memory to process the documents' contents. Therefore, this result corresponds to most of the previous work in Arabic text classification (Al-refai et al., 2007; El-halees, 2007).

Nevertheless, these pre-processing features were different according to their performance; for example, parts of speech and light stemming techniques in this project, indicated high and stable performance between all other singular features, although they were tested using two different experimental methods. On the other hand, some features such as removing stopwords and Arabic normalization indicated a fluctuating performance between sensible and insensible within different project experiments. Figures 13 and 14 show these differences. Indeed, this difference in the pre-processing feature performance was obviously in the previous work; however, the language difficulty, the classification categories complexity and the training set quality should be considered (Hmeidi et al., 2008; Nenadić et al., 2002).

Nevertheless, the light stemming technique showed a substantial improvement in performance and support for the NB classifier with the Arabic language in this project. It works to convert words into their roots to return words to their general forms by removing the very common prefixes and suffixes and keeps the word meaning unchanged. Certainly the word meaning is strongly considered in the text classification tasks. Similarly, the light stemming technique was superior on the machine learning algorithms used by Al-refai et al and Chen & Gey in their work both in classification tasks and in information retrieval in the Arabic language (Al-refai et al., 2007; Chen & Gey, 2002).

Filtering the contents of text into noun and proper noun using the parts of speech technique, as a pre-processing feature, presented substantial support for the NB classifier in this project. In the classification task, it ignores less important word classes (verb, adverb and adjective) since noun and proper noun thank highly important information within the text contents.

Actually, the part of speech technique with the purpose of including nouns and proper nouns and excluding others classes of word was used in El-halees research titled “Arabic Text Classification Using Maximum Entropy”. In this case classification performance was increased notably by around 9% (El-halees, 2007).

The collaboration of some features, to work together as one model with the learning algorithm, was really helpful. Its performance was proved by the collaborative features such as General normalization, the light stemmer and parts of speech as one model with Naïve Bayes machine learning algorithm (G1, LS, NP). It offered substantial performances through the three types of experimental method: the 10-cross validation, reverse 10-fold cross validation and normal test method in this project. They e work consecutively in order to normalize the contents of text into unification word forms and important information; hence the reduction of words domain helps the machine learning algorithm to easily generate classifier models depending on the important information, namely without noise data, in order to use this model for classifying unseen data.

5.4.2. Arabic name extraction results

Generally, Arabic the name extraction process during the CADEPI project was elementary because its goal was limited to just assisting the new dictionary performance with some pre-processing features, one of which was the part of speech technique to filter the text contents into three word classes (noun, proper noun and adjective). Essentially, it showed substantial performance without any pre-processing features and had an approximately 81% accuracy. Although, it needs more refining and editing processes in order to create a high quality Arabic name dictionary for the NLP area in the future.

Applying the rule base of Arabic names with the ANDic dictionary will provide crucial support during Arabic full name extraction. That was shown by Shaalan and Raza system, which used dictionaries and rule bases together, for person name recognition. It represented a substantial accuracy, around 89% (Shaalan & Raza 2007). However, the rule base needs further study from an Arabic linguistic view (Algahtani & McNaught, 2009).

5.5. Summary

To sum up, the pre-processing process with some features plays an essential role in the classification of Arabic contents with the NB machine learning classifier algorithm. For example, general normalization, light stemming and parts of speech. The collaborative models of pre-processing features such as (G1, LS, NP) achieved advanced accuracy in three different types of experiments (10-cross validation, reverse 10-fold cross validation and normal test).

A new experimental method was created during this project, namely the reverse 10-fold cross validation method, in order to evaluate the classification models within the limitations of the data set. It is strong test for the classification models to pass especially without examples of training sets. However, it was a fair measure because it clarified the difference between the CADEPI classification models.

The Arabic name extraction task in the CADEPI project found that ANDic was helpful as a baseline technique however it needs supportive techniques to increase its accuracy by using Arabic name rule based which should be investigated in future work.

6. Chapter 6: Conclusion and future work

The CADEPI project had two objectives: to classify Arabic documents using the Naïve Bayes classifier algorithm and full personal name extraction. The data set, which was used in the classification task, was collected, filtered and labelled manually into three categories namely economy, politics and sport. Each category consisted of 150 documents with the total number of training sets being 450. This data set was used as a training and test set using two experimental methods, the 10-fold cross validation and reverse 10-fold cross validation method. The first method is a very common investigative method in text classification task; however, the second one was created during this project so as to investigate the classifier algorithm in a limited training set and assess its performance.. A further method of investigation was applied by collecting collaborative datasets as validation data. This involved 92 documents all of which involved three categories to investigate the ideal models in this project from a practical point of view.

Five pre-processing features were applied (general normalization, Arabic normalization, removing stopwords, light stemming and parts of speech with NB classifier algorithm) in order to find more efficient features which might be used together as anideal model in the Arabic text classification task. This project observed that, these three pre-processing features (general normalization, light stemming and parts of speech) indicated substantial performance and support for the NB algorithm in the Arabic language. Therefore, grouping them as a combination of pre-processing features was a very helpful technique because this combination (G1, LS, NP) showed substantial performances and stability within the three different investigative methods with an accuracy of 99.27, 97.71, 96.7 respectively and an overall accuracy average using these methods of 97.9.

In the Arabic full name extraction, a new Arabic personal name dictionary ANDic,(which in its early stage contains 82816 Arabic person names), was used by three models. The first one used the dictionary itself (Dic) to look up and find the Arabic names and bind them as full personal names. The second model was the dictionary and parts of speech model which filtered the text contents into noun and proper noun only (Dic&NP) before the extraction process. The third model was the dictionary and parts of speech technique used together to filter the text contents this time into noun, proper noun and adjective only (Dic&NPJ) before the extraction process. When comparing the first model (Dic) outperformed all other models by 81.77 f-measure accuracy value. The dictionary indicated a good performance but it is still under the development.

One of the important implicit aims of this project was to establish and develop a standard corpus in the Arabic language involving most of the active Arabic subjects in order to be available online for research and developmental purposes particularly in NLP area.

These two objectives applied together in this project provide an integrated work that may contribute substantially to the NLP area in the Arabic language. This is the first phase of the CADEPI project because it involves many phases in which to develop a machine learning tool concerned with classification and information extraction tasks. Therefore, in the next phase extending the classification categories such as (health, science, history, religion and culture …etc) and providing more information extraction such as (birth date and phone number …etc) are planned as future goals. Additionally, further study in full Arabic person name extraction from a linguistic view to develop this current work and achieve our ambition is necessary in the next phase.

7. References

Abuleil, S. & Evens, M., 1999. Discovering Lexical Information by Tagging Arabic Newspaper Text. *Workshop on Semitic Language Processing. COLING-ACL.98, University of Montreal, Montreal, PQ, Canada*, 1-7.

Abuleil, S., 2002. Extracting Names From Arabic Text For Question-Answering Systems. *Computers and the Humanities*.

Abuleil, S., 2004. Extracting Names from Arabic Text for Question-Answering Systems. *In Proceed- ings of Coupling approaches, coupling media and coupling languages for information retrieval (RIAO 2004), Avignon, France*, (Riao), 638-647.

Al-Harbi, S. et al., 2008. Automatic Arabic Text Classification. *Journées internationales d'Analyse statistique des Données Textuelles*, 77-84.

Al-refai, M., Duwairi, R. & Khasawneh, N., 2007. Stemming Versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization Rehab Duwairil Department of Computer. *IEEE*, (September 2007), 446-450.

Al-shalabi, R. & Evens, M., 1998. A Computational Morphology System for Arabic Riyad AI-Shalabi Martha Evens Department of Computer Science and Applied Mathematics Illinois Institute of Technology 10 West 31st Street Algorithm to Find. *In Workshop on Computational Approaches to Semitic Languages, COLING-ACL98*, 66-72.

Al-shalabi, R. & Kanaan, G., 2004. Constructing An Automatic Lexicon for Arabic Language, International Journal of Computing & Information Sciences. *International Journal of Computing & Information Sciences*, 2(2), 114-128.

Al-shalabi, R., Kanaan, G. & Manaf H. Gharaibeh, 2006. Arabic Text Categorization Using kNN Algorithm. *In the Proceedings of the 4th International Multiconference on Computer Science and Information Technology (CSIT 2006), Amman, Jordan, April 5-7, 2006*, 4.

AlGahtani, S., Black, W. & Mcnaught, J., 2007. Arabic Part-Of-Speech Tagging Using Transformation-Based Learning. *In Proceeedings of the 2nd International Conference on Arabic Language Resourcesand Tools*, Cairo, Egypt, 66-70.

Algahtani, S. & McNaught, J., 2009. Improving Arabic Named Entity Recognition by Global Features and Triggers. *Proceedings of the 13th International Business Information Management Association (IBIMA)*.

Aljlayl, M. & Frieder, O., 2002. On Arabic Search : Improving the Retrieval Effectiveness via a Light Stemming Approach. *CIKM 02,McLean, Virginia, USA*, 340-347.

Ben, C., Zribi, O. & Ahmed, M.B., 2010. Efficient Automatic Correction of Misspelled Arabic Words Based on Contextual Information. *Computational Linguistics*, 770-777.

Chen, A. & Gey, F., 2002. Building an Arabic Stemmer for Information Retrieval. *In Proceedings of the 11th Text Retrieval Conference (TREC 2002), National Institute of Standards and Technology*.

Davis, J. & Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd international conference on Machine learning - ICML '06*, 233-240. Available at: http://portal.acm.org/citation.cfm?doid=1143844.1143874.

El-halees, A.M., 2007. Arabic Text Classification Using Maximum Entropy. *The Islamic University Journal (Series of Natural Studies and Engineering)*, 15(1), 157-167. Available at: http//www.iugzaza.edu.ps/ara/research/.

El-halees, A.M., 2006. Mining Arabic Association Rules for Text Classification. *In the proceedings of the first international conference on Mathematical Sciences. Al-Azhar University of Gaza, Palestine*, 15-17.

Elkourdi, M., Bensaid, A. & Rachidi, T., 2004. Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. *20th International Conference on Computational Linguistics August, Geneva*.

Elsebai, A., 2009. A Rule Based Persons Names Arabic Extraction System. *System*, 11, 53-59.

Halpern, J., 2009. Lexicon-Driven Approach to the Recognition of Arabic Named Entities. *Proceedings of the Second International Conference on Arabic Language Resources and Tools. Cairo: MEDAR.*

Hmeidi, I., Hawashin, B. & El-qawasmeh, E., 2008. Performance of KNN and SVM classifiers on full word Arabic articles. *Advanced Engineering Informatics*, 22, 106-111.

Khoja, S., 2001. APT : Arabic Part-of-speech Tagger. *Proc. of the Student Workshop at NAACL 2001*, 20-26.

Khreisat, L., 2009. A machine learning approach for Arabic text classification using N-gram frequency statistics. *Journal of Informetrics*, 3, 72-77. Available at: www.elsevier.com/locate/joi.

Larkey, L.S. et al., 2002. Improving Stemming for Arabic Information Retrieval : Light Stemming and Co-occurrence Analysis. *SIGIR'02, August 11-15, 2002, Tampere, Finland.*, 275-282.

Lewis, D.D., 1994. A Comparison of Two Learning Algorithms for Text Categorization 1 Introduction 2 Text Categorization : Nature and Approaches. *Nature*, 1-14.

Maamouri, M. et al., 2004. The Penn Arabic Treebank : Building a Large-Scale Annotated Arabic Corpus. *In NEMLAR Conference on Arabic Language Resources and Tools, Cairo,Egypt.*

Maloney, J., Niv, M. & Corp, S.R., 1988. TAGARAB : A Fast , Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. *In Proceedings of the Workshop on Computational Approaches to Semitic Languages*, 8-15.

Mccallum, A. & Nigam, K., 1998. A Comparison of Event Models for Naive Bayes Text Classification. *In AAAI-98 Workshop on Learning for Text Categorization*, 4148.

Mesfar, S., 2007. Named Entity Recognition for Arabic Using Syntactic. *In: Natural Language Processing and Information Systems: Springer Berlin / Heidelberg*, 305-316.

Mitchell, T., 1997. *Machine Learning* internatio., McGraw-Hill.

Nenadić, G. et al., 2002. Selecting Text Features for Gene Name Classification : from Documents to Terms. *Science*, 121-128.

Refaeilzadeh, P., Tang, L. & Liu, H., 2008. Cross-Validation. *Arizona State University*.

Sakhr, 2010. text anaylesis. *Sakhr*. Available at: http://textmining.sakhr.com/Default.aspx.

Savoy, J., 2010. IR Multilingual Resources at UniNE. *University of Neuchatel*. Available at: http://members.unine.ch/jacques.savoy/clef/.

Sawaf, H., Ney, H. & Ag, A., 2001. Statistical Classification Methods for Arabic News Articles. *Natural Language Processing in ACL2001, Toulouse, France.*

Shaalan, K. & Raza, H., 2007. Person Name Entity Recognition for Arabic. *Proceedings of the 5th Workshop on Important Unresolved Matters, pages 17–24, Prague, Czech Republic, June 2007*, (June), 17-24.

Sokolova, M. & Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4), 427-437. Available at: http://dx.doi.org/10.1016/j.ipm.2009.03.002.

Soon, W.M. et al., 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4), 521-544.

Syiam, M.M., Fayed, Z.T. & Habib, M.B., 2006. AN INTELLIGENT SYSTEM FOR ARABIC TEXT CATEGORIZATION. *IJICIS*, 6(1), 1-19.

Toutanova, K. & Manning, C.D., 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *In EMNLP/VLC 1999*, 63-71.

Yang, Y. & Liu, X., 1999. A re-examination of text categorization methods. *In Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 42-49.

Zerrouki, T. & Amara, M., 2009. Arabic Stop words. *sourceforge*. Available at: http://sourceforge.net/projects/arabicstopwords/files/.

Collection data source

http://www.sama.gov.sa/Pages/Home.aspx.
http://www.cma.org.sa/cma_en/default.aspx
http://www.aljazeera.net/portal.
http://www.alriyadh.com/.
http://www.aleqt.com/.
http://www.alwatan.com.sa.
http://arabic-media.com/newspapers/egypt/ahram.htm

8. Appendices: source code

These parts of CAFEPI source codes just to highlight the important parts.

- ***General normalization process (G1).***

```
   static String Normalaization1(String word)
{
    int len = word.length(); //strlen (word)-1;
    String word1;

    word1 = word;
    if(len > 2)
    {
           if (word.endsWith("."))
           { word =  word.substring(0,len-1);}

            if (word.startsWith("."))
           { word =  word.replace(".","");}

             if (word.endsWith(".."))
           { word =  word.substring(0,len-2);}

            if (word.startsWith(".."))
           { word =  word.replace("..","");}

            if (word.endsWith(","))
           { word =  word.substring(0,len-1);}

            if (word.startsWith(","))
           { word =  word.replace(",","");}

            if (word.endsWith(";"))
           { word =  word.substring(0,len-1);}

            if (word.startsWith(";"))
           { word =  word.replace(";","");}

            if (word.endsWith(":"))
           { word =  word.substring(0,len-1);}

            if (word.startsWith(":"))
           { word =  word.replace(":","");}

            if (word.endsWith("("))
           { word =  word.substring(0,len-1);}

            if (word.startsWith(")"))
           { word =  word.replace(")","");}

            if (word.endsWith("'"))
           { word =  word.substring(0,len-1);}

            if (word.startsWith("'"))
           { word =  word.replace("'","");}

            if (word.endsWith("\""))
           { word =  word.substring(0,len-1);}

            if (word.startsWith("\""))
           { word =  word.replace("\"","");}

            if (word.endsWith("?"))
           { word =  word.substring(0,len-1);}

            if (word.startsWith("?"))
           { word =  word.replace("?","");}

            if (word.endsWith("!"))
           { word =  word.substring(0,len-1);}

            if (word.startsWith("!"))
           { word =  word.replace("!","");}

           if (word.endsWith("،"))
           { word =  word.replace("،","");}

            if (word.startsWith("،"))
           { word =  word.replace("،","");}
    }

    return word;

}
```

- ***Arabic normalization process (G2).***

```
static String Normalaization2(String word)
{
    int len = word.length(); //strlen (word)-1;
    String word1;

    word1 = word;

    if (word.endsWith("ى"))
   {  word = word.replace("ى","ي");}

    if (word.endsWith("ة"))
   { word = word.replace("ة","ه");}

    if (word.startsWith("أ"))
  {  word = word.replace("أ","ا");}

    if (word.startsWith("إ"))
  {  word = word.replace("إ","ا");}

    if (word.startsWith("آ"))
  { word =  word.replace("آ","ا");}

    return word;

}
```

- ***Light stemming process (removing very common prefix and suffix).***

```
static String light_stemmer(String fileCntents)
{

        StringTokenizer FileToken = new
StringTokenizer(fileCntents, " \t\n\r\f");
        String word="",Readyfilecontents ="";

                while (FileToken.hasMoreTokens())
                {
                    word = FileToken.nextToken();

                    if(!IsItStopword(word))
                    {
                        word = ArabicPrefix(word);
                        word = ArabicSuffix(word);
                        //System.out.println("After light stemmer
"+word);
                        Readyfilecontents = Readyfilecontents + " "
+ word;
                    }

                }

        return Readyfilecontents;

}
```

- ***Removing Arabic prefix process.***

```
static String ArabicPrefix(String word)
{

    String wordorg;

    wordorg = word;


    if (word.startsWith("لل"))
   {    word =  word.replace("لل",""); }

    if (word.startsWith("ال"))
   { word =  word.replace("ال","");}

    if (word.startsWith("كال"))
   { word =  word.replace("كال",""); }

    if (word.startsWith("فال"))
   { word =  word.replace("فال",""); }

    if (word.startsWith("بال"))
   { word =  word.replace("بال","");}

     if (word.startsWith("و"))
   { word =  word.replace("و","");}


    if(word.length()>=3)
    { return word;
    }
    else return wordorg;
}
```

- ***Removing Arabic suffix process.***

```
static String ArabicSuffix(String word)
{

    String wordorg;

    wordorg = word;


    if (word.endsWith("هما"))
   {    word =  word.replace("هما",""); }

    if (word.endsWith("كـما"))
   { word =  word.replace("كـما","");}

    if (word.endsWith("ات"))
   { word =  word.replace("ات",""); }

     if (word.endsWith("يـه"))
   { word =  word.replace("يـه",""); }

   if (word.endsWith("تـه"))
   { word =  word.replace("تـه","");}

     if (word.endsWith("تـي"))
   { word =  word.replace("تـي","");}

    if (word.endsWith("ان"))
   { word =  word.replace("ان","");}

      if (word.endsWith("ون"))
   { word =  word.replace("ون","");}

    if (word.endsWith("يـن"))
   { word =  word.replace("يـن","");}

    if (word.endsWith("هم"))
   { word =  word.replace("هم","");}

    if (word.endsWith("هن"))
   { word =  word.replace("هن","");}
```

```
     if (word.endsWith("هـا"))
   { word =  word.replace("هـا","");}

     if (word.endsWith("نـا"))
   { word =  word.replace("نـا","");}

     if (word.endsWith("وا"))
   { word =  word.replace("وا","");}

    if (word.endsWith("كم"))
   { word =  word.replace("كم","");}

    if (word.endsWith("كـن"))
   { word =  word.replace("كـن","");}

    if (word.endsWith("ي"))
   { word =  word.replace("ي","");}

     if (word.endsWith("ــه"))
   { word =  word.replace("ــه","");}

    if(word.length()>3)
    { return word;
    }
    else return wordorg;
}


   static boolean IsItStopword(String word)
   {

       return ArabicStopwordsList.containsKey(word);

    }


   static boolean IsItNN(String word)
   {

       return ArabicNNList.containsKey(word);

    }
```

- ***Using the out put of TBL post tagger to extract noun and proper noun process.***

```
   static String RemovingUnNNJJ(String fileCntents)
   {

        StringTokenizer FileToken = new
StringTokenizer(fileCntents, " \t\n\r\f");
        String word="",Readyfilecontents ="";

                while (FileToken.hasMoreTokens())
                {
                    word = FileToken.nextToken();

                    if(IsItNN(word))
                    {
                       Readyfilecontents = Readyfilecontents + " "
+ word;
                          // System.out.println(word);
                    }
                }

        return Readyfilecontents;
    }


   static String ArabicNormalizition(String fileCntents)
   {

        StringTokenizer FileToken = new
StringTokenizer(fileCntents, " \t\n\r\f");
        String word="",Readyfilecontents ="";

                while (FileToken.hasMoreTokens())
                {
                    word = FileToken.nextToken();

                    word = Normalaization2(word);

                    Readyfilecontents = Readyfilecontents + " " +
word;
                }

        return Readyfilecontents;
    }

   static String GeneralNormalizition(String fileCntents)
   {

        StringTokenizer FileToken = new
StringTokenizer(fileCntents, " \t\n\r\f");
        String word="",Readyfilecontents ="";

                while (FileToken.hasMoreTokens())
                {
                    word = FileToken.nextToken();

                    word = Normalaization1(word);

                    Readyfilecontents = Readyfilecontents + " " +
word;
                }

        return Readyfilecontents;
    }
```

- ***Removing Arabic stopwords process.***

```
   static String RemovingStopWords(String fileCntents)
   {

        StringTokenizer FileToken = new
StringTokenizer(fileCntents, " \t\n\r\f");
        String word="",Readyfilecontents ="";

                while (FileToken.hasMoreTokens())
                {
                    word = FileToken.nextToken();

                    if(!IsItStopword(word))
                    {
                        Readyfilecontents = Readyfilecontents + " "
+ word;
                    }

                }
        return Readyfilecontents;
    }
```

- ***NaiveBayesClassifier process***

```
   static String NaiveBayesClassifiers(String testfileContent)
  {

       StringTokenizer testFileToken = new
StringTokenizer(testfileContent, " \t\n\r\f");
        String word="";

        int wocear=0;
       double  EcoValue=1,EcoValueT=0,PolValue =1,PolValueT =0,
SprValue=1,SprValueT=0,numerator=0,denominator=0;


        pcEcofile = (NumOfEconomyFiels/NumOfAllFiles);
        pcPolfile = (NumOfPoliticFiles/NumOfAllFiles);
        pcSprfile = (NumOfSportFiles/NumOfAllFiles);


       while (testFileToken.hasMoreTokens())
       {
           word = testFileToken.nextToken();

           if(All.containsKey(word))
            {

                 numerator =0; denominator =0; wocear = 0;
                    if(Eco.containsKey(word))
                    {
                        wocear = Eco.get(word);
                    }
                    else
                    {
                        wocear = 0;
                    }

                        numerator = wocear + 1;// anomiator
                        denominator = EconomyWordsCounter +
Vocabulry;

                        EcoValueT = numerator/denominator;
                        EcoValue = EcoValue +
Math.log(EcoValueT);


                 numerator =0; denominator =0; wocear = 0;
                 if(Pol.containsKey(word))
                 {
                    wocear = Pol.get(word);
                 }
                 else
                 {
                     wocear = 0;
                 }

                     numerator = wocear + 1;
                     denominator = PoliticWordsCounter +
Vocabulry;

                     PolValueT = numerator/denominator;
                     PolValue = PolValue +
Math.log(PolValueT);


                 numerator =0; denominator =0; wocear = 0;
                 if(Spr.containsKey(word))
                 {
                    wocear = Spr.get(word);
                 }
                 else
                 {
                     wocear = 0;
                 }

                     numerator = wocear + 1;
                     denominator = SportWordsCounter +
Vocabulry;

                     SprValueT = numerator/denominator;
                     SprValue = SprValue +
Math.log(SprValueT);

            }
       }
                //--------------with log-----------------------
-----

                EcoValue = Math.log(pcEcofile)+ EcoValue;
                PolValue = Math.log(pcPolfile)+ PolValue;
                    SprValue = Math.log(pcSprfile)+ SprValue;
```

```
               //---------------------------------------------
------

     if (PolValue > EcoValue && PolValue > SprValue )
     {
          return "Politic";
                    }
      else if (EcoValue > PolValue && EcoValue > SprValue )
     {
          return "Economy";
                    }
      else if (SprValue > PolValue && SprValue > EcoValue )
      {
            return "Sport";
      }
      else
      {
            return "Noun classfied";
      }
}


static void NameRecognation2(String testfileContent,String
filename)
  {

   StringTokenizer testFileToken = new
StringTokenizer(testfileContent, " \t\n\r\f");

   String fullName =""; String word ="";
   int cnt =0; int NoNames =0;

   while (testFileToken.hasMoreTokens())
     {
          word = testFileToken.nextToken();
            word = Normalaization1(word);

            //if(!IsItStopword(word) &&
ArabicNamesList.containsKey(word))
            if(!IsItStopword(word) &&
ArabicNamesList.containsKey(word) && IsItNN(word))
            {
               fullName = fullName + " " + word;
               cnt++;
             }

           else
            {
              if(cnt >1)
              {
                 NoNames++;
                System.out.println("| "+NoNames + " - " +
fullName);
                Resultout.println("| "+NoNames + " - " +
fullName);
              }
              fullName ="";
              cnt =0;
           }

     }
}
```

- ***Main function for CADEPI project process.***

```
  public static void main(String[] args)
  {


        // testset  trainingset   trainingset1-10-315T
testset-out
        File CurrDir = new File ("CollaborativeDataSet");
        File[] files = CurrDir.listFiles();

        //String TriningSetDir = "trainingset-1";
       String Likelihood = "";
        String TestFileContent="";

       LoadingArabicStopwordsList("ArabicStopWords.txt");
        LoadingArabicNNList("NP.txt");
        LoadingArabicNamesList("ArbName.txt");


        TrainingSet("trainingset-150");
try{
        outFile = new
FileWriter("CADEPI_"+CurrDir+"_G1LSNP.txt");
        Resultout = new PrintWriter(outFile);
```

```
           System.out.println("|------------Training set
information ----------------");
           System.out.println("| Class    -    Files No -  Words No
");
           System.out.println("| Economy  -    " +
NumOfEconomyFiels +"       "+ EconomyWordsCounter);
           System.out.println("| Politics -    " +
NumOfPoliticFiles +"       "+ PoliticWordsCounter);
           System.out.println("| Sport    -    " + NumOfSportFiles
+"       "+  SportWordsCounter);
           System.out.println("|-----------------------------------
-----------------");
           System.out.println("| Vocabulary No :           "+
Vocabulry);
           System.out.println("|-----------------------------------
-----------------");



           Resultout.println("|------------Training set information
-----------------");
           Resultout.println("| Class    -    Files No -  Words No
");
           Resultout.println("| Economy  -    " + NumOfEconomyFiels
+"       "+ EconomyWordsCounter);
           Resultout.println("| Politics -    " + NumOfPoliticFiles
+"       "+ PoliticWordsCounter);
           Resultout.println("| Sport    -    " + NumOfSportFiles
+"       "+  SportWordsCounter);
           Resultout.println("|------------------------------------
----------------");
           Resultout.println("| Vocabulary No :           "+
Vocabulry);
           Resultout.println("|------------------------------------
----------------");



           double error =0; int TEcofiles =0; int TPolfiles =0; int
TSprfiles =0;
           double Ecoerror =0; double Polerror =0; double Sprerror
=0;


                for(File file : files)
```

```
              {
                    String fileName = file.getName();

                        TestFileContent = FileContentsWPO(CurrDir +
"/" + fileName);

                        Likelihood =
NaiveBayesClassifiers(TestFileContent);
                        System.out.println("|This file ("+ fileName
+") clasified as:" + Likelihood);
                        Resultout.println("|This file ("+ fileName
+") clasified as:" + Likelihood);

                        // ----statistic calculation-----

                        if(fileName.contains("Economy"))
                        {
                            TEcofiles++;
                        }

                         if(fileName.contains("Economy") &&
!Likelihood.contains("Economy"))
                        {
                            Ecoerror++;
                        }

                         if(fileName.contains("Politic"))
                        {
                            TPolfiles++;
                        }

                         if(fileName.contains("Politic") &&
!Likelihood.contains("Politic"))
                        {
                            Polerror++;
                        }

                       if(fileName.contains("Sport"))
                        {
                            TSprfiles++;
                        }

                         if(fileName.contains("Sport") &&
!Likelihood.contains("Sport"))
                        {
                            Sprerror++;
                        }

                        if(!fileName.contains(Likelihood))
                        {
                            error++;
                        }


                         TestFileContent = FileContentsB(CurrDir +
"/" + fileName);

//NameRecognation2(TestFileContent,fileName);


                        System.out.println("|----------------------
------------------------------------|");
                }
            //out.println("----------------------------------------
---");



            System.out.println("|------------Test set information -
------------------");
            System.out.println("|class    - " +" File No - "+ "
Error - " + " Accuracy  ");
            System.out.println("|Economy  -    " +TEcofiles+"
" + Ecoerror+"      "    + ((TEcofiles-Ecoerror)/TEcofiles)*100 );
            System.out.println("|Politics -    " +TPolfiles+"
" + Polerror+"      "    + ((TPolfiles-Polerror)/TPolfiles)*100 );
            System.out.println("|Sport    -    " +TSprfiles+"
" + Sprerror+"      "    +((TSprfiles-Sprerror)/TSprfiles)*100 );
            System.out.println("|----------------------------------
------------------");
            int testfiles = TSprfiles + TSprfiles + TPolfiles +
TEcofiles;
            System.out.println("|Final result:           "+ error
+"      "+ ((testfiles-error)/testfiles)*100 );
            System.out.println("|----------------------------------
------------------");


            Resultout.println("|------------Test set information --
------------------");
            Resultout.println("|class    - " +" File No - "+ "
Error - " + " Accuracy  ");
```

```
            Resultout.println("|Economy  -    " +TEcofiles+"
" + Ecoerror+"      "    + ((TEcofiles-Ecoerror)/TEcofiles)*100 );
            Resultout.println("|Politics -    " +TPolfiles+"
" + Polerror+"      "    + ((TPolfiles-Polerror)/TPolfiles)*100 );
            Resultout.println("|Sport    -    " +TSprfiles+"
" + Sprerror+"      "    +((TSprfiles-Sprerror)/TSprfiles)*100 );
            Resultout.println("|-----------------------------------
-----------------");
            Resultout.println("|Final result:           "+ error +"
"+ ((testfiles-error)/testfiles)*100 );
            Resultout.println("|-----------------------------------
-----------------");

            Resultout.close();

    }catch (IOException e){System.out.println("IOException : " +
e);}


   }

}
```