Université de Blida 1–Saad Dahlab

Faculté des sciences

Département d'Informatique

Mémoire présenté par :

Mlles. ZIBANI Seloua et ZOUID Sirine

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Traitement Automatique de la Langue

Sujet:

Arabic Poems Classification According to their Eras and Topics

Soutenu le : 23/09/2020, devant le jury composé de :

M. HAMOUDA	Université de Blida 1	Président
Mme. CHERGUENE	Université de Blida 1	Examinatrice
M. M. ABBAS	CRSTDLA	Encadreur
Mme. M. MEZZI	Université de Blida 1	Promotrice

Abstract

Arabic text classification has faced many difficulties because of the Arabic language natural especially poetry which is very challenging form of the Arabic text. The big growth of the Arabic internet content in the last years has raised up the need for Arabic text processing tools because of its unstructured data. And because poetry classification didn't get the attention that it deserves, it becomes one of our motivations for this work.

The lack of open source corpora, lead us to build two corpora 'Adab' and 'Aldiwan' of Arabic poetry. We used these two corpora to build a system of Arabic Poems Classification according to their category, era and topic.

We used machine learning techniques in which we applied a numerous methods and classifiers. The best results were obtained using the LSVC (Linear Support Vector Classification) algorithm. The last model that we obtained in our experiments, we used it to build our application which we called "Saline Classification".

Keywords: Poetry Classification, Arabic Language, Category Identification, Era Identification, Topic Identification, Machine Learning

Résumé

La classification des textes arabes a rencontré de nombreuses difficultés en raison de la langue arabe naturelle, en particulier la poésie qui est une forme très difficile du texte arabe. La forte croissance du contenu arabe sur Internet ces dernières années a fait naître le besoin d'outils de traitement de texte arabe en raison des données non structurées. Et parce que la classification de la poésie n'a pas reçu l'attention qu'elle mérite, c'est devenu une de nos motivations pour ce travail.

L'absence de corpus open source, nous a conduits à construire deux corpus, "Adab" et "Aldiwan", de poésie arabe. Nous avons utilisé ces deux corpus pour construire un système de classification des poèmes arabes selon leur catégorie, leur époque et leur sujet.

Nous avons utilisé des techniques d'apprentissage automatique dans lesquelles nous avons appliqué de nombreuses méthodes et classificateurs. Les meilleurs résultats ont été obtenus en utilisant l'algorithme LSVC (Linear Support Vector Classification). Le dernier modèle que nous avons obtenu lors de nos expériences, nous l'avons utilisé pour construire notre application que nous avons appelée "Saline Classification".

Mots-clés : Classification de la poésie, Langue arabe, Identification de la catégorie, Identification de l'époque, Identification du sujet, Apprentissage automatique.

ملخص

واجــه تصــنيف الــنص العربــي العديــد مــن الصــعوبات بســبب اللغــة العربيــة الطبيعيــة وخاصــة الشــعر الــذي يمثــل تحــديًا كبيــرًا للــنص العربــي. أدى النمــو الكبيــر لمحتــوى الإنترنــت العربــي فــي الســنوات الأخيــرة إلــى زيــادة الحاجــة إلــى أدوات معالجــة النصــوص العربيــة بســبب بيانــات اغيــر منظمــة. ولأن تصــنيف الشــعر لــم يحــظ بالاهتمــام الذي يستحقه ، فقد أصبح أحد دوافعنا لهذا العمل.

أدى نقــص المصــادر المفتوحــة إلــى بنــاء مجمــوعتين "أدب" و "الــديوان" مــن الشــعر العربــي. اســتخدمنا هــاتين المجمــوعتين لبنــاء نظــام لتصــنيف القصــائد العربيــة حســب الفئة والعصر والغرض الشعري.

اســتخدمنا تقنيــات الــتعلم الآلــي التــي طبقنــا فيهــا العديــد مــن الأســاليب والمصــنفات. تــم الحصــول علــى أفضــل النتــائج باســتخدام خوارزميــة LSVC (تصــنيف ناقــل الــدعم الخطــي). آخــر نمــوذج حصــلنا عليــه فــي تجاربنــا ، اســتخدمناه لبنــاء تطبيقنــا الــذي أطلقنــا عليه اسم "SALINE Classification"

الكلمــات المفتاحيــة: تصــنيف الشــعر ، اللغــة العربيــة ، تحديــد الفئــة ، تحديــد العصــر ، تحديد الغرض الشعري، التعلم الآلي.

Acknowledgment

First and foremost, praises and thanks to the God, the Almighty, for his showers of blessings to complete our work successfully. Also we wish to thank our Family for their support and encouragement throughout our study.

We would like to express our sincere gratitude to our advisor Mrs. Mezzi for the continuous support, patience, motivation, and enthusiasm. Her guidance helped us in all the time of research and writing of this thesis. We could not have imagined having a better advisor and mentor.

Besides our advisor, we would like to thank Mr. Abbas for offering us the internship opportunity in their Center and leading us working on diverse exciting project and put faith in us to realize it.

Our sincere thanks also goes to Mr. Lichouri, a researcher in CRSTDLA Center for his effort and immense knowledge. He has been a great help for us to realize this work.

Dedication

Every challenging work needs self-effort, determination as well as guidance of elders especially those who were very close to our hearts.

I dedicate my humble effort to my sweet and loving

Parents,

A strong and gentle souls who taught me to trust Allah, believe in hard work and that so much could be done with little and nothing is possible. They showed me the way of success and live my life according to my rules.

Siblings,

Shiraz, Faiz & Rayane (aka Divi) who never left my side, for their illumined support, honest opinion, feedback and generous faith in me. Moreover, their endless love.

Also my Friends, your advices, encouragements and wishing me the best of luck, it means a lot to me.

Thank You …Thank You All Without I'm Nothing… You Made Who I am Today.

In a world full of lies, you are the truth to me, and no matter how much hard life can get there's always a way to change the game.

Sirine

Dedication

This study is wholeheartedly dedicated to my "Mom" who have been my source of strength and continually provide her moral, spiritual, emotional and financial support.

To "Dad" my God have mercy on him and make his resting place in Paradise which I know he will be proud of me if he was in here with me.

To my four brothers Amine, Hichem, Mohamed and Bilel who supported me throughout my study process.

To my sisters from deferent mothers Romayssa, Meriem, Manel, Sabrina (Bougy) and Amina who never left my side with a special feeling of gratitude to be in my life.

Also, to all my family members who shared their words of advice and encouragement to finish this study.

And lastly, to Almighty God for the guidance, strength, power, protection, and skills.

Saloua

Table of Contents

List of Figures

List of Tables

List of Abbreviation

AD: Arabic Dialect

ADAM: Analyzer for Dialectal Arabic Morphology

ANLP: Arabic Natural Language Processing

APC: Arabic Poem Classification

ATC: Arabic Text Classification

BNB: Bernoulli Naïve Bayes

CA: Classical Arabic

DT: Decision Tree

GB: Gradient Boosting

HTML: Hypertext Markup Language

IR: Information Retrieval

KNN: K Nearest Neighbor

LR: Logistic Regression

LSTM: Long Short-Term Memory

LSVC: Linear Support Vector Classification

MFCC: Mel Frequency Cepstral Coefficient

MNB: Multinomial Naïve Bayes

MSA: Modern Standard Arabic

MT: Machine Translate

NB: Naïve Bayes

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

NN: Neural Network

PATB: Penn Arabic Treebank

POS: Part-of-speech

QAC: Quranic Arabic Corpus

RE: Regular Expression

RF: Random Forest

RNN: Recurrent Neural Network

SAS: Stanford Arabic Segmenter

SVM: Support Vector Machine

TC: Text Classification

TD: Tunisian dialect

UTF-8: Universal Character Set Transformation Format - 8 bits

WSD: Word Sense Disambiguation

General Introduction

1. General context

Despite the rapid progress in the analysis of poetry in some international languages, the analysis of the Arabic poetry has not received a sufficient attention due to the difficulty of the Arabic language, and the difficulties of analyzing its poetic theories.

The "DAKHIRA AL Arabiya" project, is an initiative of the Arab league based on the principle of participation of scientific and cultural institutions of each member country, including the Scientific and Technical Research Center for the Development of the Arabic Language (CRSTDLA) which contributes to this bold idea which is a database of ancient and modern textual data covering the Arab cultural heritage.

2. Research Problems

The classification of Arabic poetry has its own difficulties and limitations resulting from the nature of the Arabic language which is a language rich in varieties with a very complex morphology which can make an ordinary analysis a very complicated task.

Lack of publicly available Arabic corpus for evaluating our system has been a major problem of our research. Many researcher to test and evaluate their system need to create their own corpus and we did the same.

3. Research Challenges

The main objectives of this work is to:

- Propose a system that classify Arabic poetry according to their eras and topics.
- Creating corpora that deals with Arabic poetry.
- Deals with poems in their correcting forms free (الشعر الحر) and not free poems (الشعر العمودي).

4. Thesis Organization

Our thesis consists of general introduction, general conclusion and five chapters that are organized as follows:

- Chapter 1 “Arabic Natural Language Processing”, here we talked about the Arabic Language and its proprieties. Then, we spook about the Arabic natural language processing its challenges, objectives, problems and works.
- Chapter 2 “Text Classification”, here we talked about text classification in general, its process, types, problems and application. Then, about the Arabic text classification and its works.
- Chapter 3 “Arabic Poems Classification”, here we took a tour in the Arabic poetry, its importance, types, eras and topics. Then, we talked about Arabic poems classification, its works and difficulties.
- Chapter 4 “Conception”, here we talked about the global architecture of our system. Then, we spook about each part of it in details.
- Chapter 5 “Experiments and Implementation”, here we explained the experiments and the obtained results along with the implementation of our application.

Chapter I:

Arabic Natural Language Processing

1. Introduction

Language is a system of conventional spoken, manual (signed), or written symbols by means of which human beings, as members of a social group and participants in its culture, express themselves. The functions of language include communication, the expression of identity, play, imaginative expression, and emotional release [1].

The interest in language from the developer's perspective started at the very beginning of software engineering, particularly in the field of Artificial Intelligence; then we witnessed the birth of the NLP (Natural Language Processing).

During the 1990s, the pace of development of NLP expanded. Language structures, instruments, and functional assets identified with NLP opened up with the parsers. The exploration of the center and advanced themes, for example, word sense disambiguation and factually hued NLP, the work on the dictionary got a heading of research. This journey for the development of NLP was joined by other fundamental themes, for example, measurable language handling, Information Extraction, and programmed summarizing.

In this chapter, we will quickly address the Arabic Natural Language Processing while portraying particularities of the Arabic Language just as its morphology and syntactic properties.

2. Arabic Natural Language Processing

Arabic Natural Language Processing (ANLP) is a growing discipline where there is an expanding number of research and technologies that focus on the specificities of this language and provide the essential tools for the development of its processing. The ANLP is a difficult field of research. It contains challenges and complex issues relevant to the morphological multifaceted nature of the Arabic language, its ambiguity, and the presence of numerous dialects with critical variations.

2.1. Arabic Natural Language Processing Challenges

Over the last few years, Arabic natural language processing (ANLP) knew an exponential growth and gained an increasing importance. ALNP applications must deal with several complex problems pertinent to the nature and structure of the Arabic language. The most well-known challenges are [2]:

- Arabic is written from right to left.
- There is no capitalization in Arabic.
- Arabic letters change shape according to their position in the word.
- Modern Standard Arabic (MSA) does not have orthographic representation of short letters which requires a high degree of homograph resolution and word sense disambiguation.
- Arabic is a pro-drop language, it allows subject pronouns to drop subject to recoverability of deletion.
- Arabic texts include many translated and transliterated named entities whose spelling, in general, tends to be inconsistent in Arabic texts. For example a named entity such as the city of Washington could be spelled ‘واشنجطن’ ،‘واشنغطن’ ،‘واشنطن’ ،‘وشنطن’.
- The lack of a sizable corpus of Arabic named entities, which would have helped both in rule-based and statistical named entity recognition systems.

2.2. Arabic Natural Language Processing Objectives

ANLP tools that could scan such documents to recognize names, places, dates, etc., of interest soon became essential. As a result, funding became available for companies and research centers to develop tools such as named entity recognition, machine translation, especially spoken machine translation, document categorization. On the other hand, ANLP applications developed in the Arab World have different objectives and usually employ

both rule-based and machine-learning approaches. The following are some of the objectives of ANLP for the Arab World [2]:

1. Transfer of knowledge and technology to the Arab World. Most recent publications in science and technology are published in the English language and are not accessible to Arab readers with little or no competence in English. To use human translators to translate such an enormous amount of data to Arabic is very costly and time consuming. So Arabic NLP could help reduce the time and cost of translating, summarizing, and retrieving information in Arabic for Arab speakers.
2. Modernize and fertilize the Arabic language. This follows from (1) above. Translating new concepts and terminology into Arabic involves coinage, arabization, and making use of lexical gaps in the Arabic language. This will positively affect the revitalization of the Arabic language and enable it to fulfill the essential needs for its speakers.
3. Improve and modernize Arabic linguistics. Arabic NLP needs a more formal and precise grammar of Arabic than the traditional grammar so widely employed today. Innovation is needed as well to preserve the valuable heritage of traditional Arab grammarians.
4. Make information retrieval, extraction, summarization, and translation available to the Arab user. The hope is to bridge the gap between peoples of the Arab world and their peers in more technically advanced countries. By making information available to Arabic speakers in their native language, Arabic NLP tools empower the present generation of educated Arabs. Thus Arabic NLP tools are indispensable in the struggle of Arabic speakers to attain parity with the rest of the world which is, in turn a matter of national security to the Arab World.

The field of Natural Language Processing applied to the Arabic language has gained significant progress throughout the years. However there is still a long way to go to be able to compete with other languages such as English and French.

3. The Arabic Language

Arabic is one of the six official UN languages, it is recognized as the 4th most used language of the Internet. It consider as Semitic language spoken by more than 330 million people as a native language, in an area extending from the Arabian/Persian Gulf in the East to the Atlantic Ocean in the West. It is rooted in the Classical or Quranic Arabic, but over the centuries, the language has created to what is presently acknowledged as MSA [3].

The Arabic language is both challenging and interesting. It is interesting due to its history the strategic importance of its people and the region they occupy, and its cultural and literary heritage. It is also challenging because of its complex linguistic structure.

At the historical level, Classical Arabic has remained unchanged, intelligible and functional for more than fifteen centuries. Culturally, the Arabic language is closely associated with Islam and with a highly esteemed body of literature. Strategically, it is the native language of more than 330 million speakers living in an important region with huge oil reserves crucial to the world economy, and home as well to the sacred sites of the world three monotheistic religions. It is also the language in which 1.4 billion Muslims perform their prayers five times a day. Linguistically, it is characterized by a complex Diglossia[1] situation. Chronologically Classical Arabic represents the language spoken by the Arabs more than fourteen centuries ago, while Modern Standard Arabic is an evolving variety of Arabic with constant borrowings and innovations proving that Arabic reinvents itself to meet the changing needs of its speakers. At the regional level there are as many Arab dialects as there are members of the Arab

[1] A phenomenon whereby two or more varieties of the same language exist side-by-side in the same speech community

league. Despite its cultural, religious, and political significance, Arabic has received comparatively little attention in modern computational linguistics [2].

3.1. Arabic Particularities

The Arabic language is written from right to left and it has 28 letters (29 if we included the “hamza” as a letter), some of which have one structure ("د"), while others have two structures ("س"; "ـسـ"), three structures (" ;"ـهـ ه"; "ـهـ"") or four structures عـ"; "ـعـ"; "ـع"–"] [ع"; " ;the notions of capital letters and lower case letters do not exist.

3.1.1. Voyellation

An Arabic lexical unit is written with consonants and vowels. Vowels are added above or below the letters. They are necessary for the correct reading and comprehension of a text and they allow us to differentiate between lexical units with the same representation. For a better understanding, let us take the example مدرسة of Table 1-1.

Table 1-1: Ambiguity caused by the absence of vowels for lexical units’ حمدرسة.

Lexical Unite	1st interpretation	2nd interpretation
مدرسة	school مَدرَسة	teacher مُدَرّسة

3.1.2. Agglutination

Arabic shows a strong tendency to agglutinate: all the morphemes stuck together and constituting a lexical unit convey several morpho-syntactic information. These lexical units can often be translated by the equivalent of a sentence in French. Example: The Arabic word “أتتـذكروننا" corresponds in English to the phrase "Do you remember us?".

3.1.3. Word Order

In Arabic, the word to which you want to draw attention or the word with the most meaning is placed at the beginning of the sentence. This order causes artificial syntactic ambiguities in that the grammar must include all the rules for possible combinations of word order inversion in the sentence. Thus, for example, one can change the order of sentence (1) to obtain another sentence (2) with the same meaning [4].

1) تعلم الطفل القراءة والكتابة في المدرسة . / the child has learned to read and write at school.

2) في المدرسة تعلم الطفل القراءة و الكتابة . / at school the child has learned to read and write.

3.2. Arabic Morphology

Arabic is a highly structured and derivational language with a rich and complex morphology that plays an important role.

As a branch of linguistics, morphology is the study of the internal structure of words or the way in which words are constructed out of smaller meaningful units. Word forms can often be analyzed into segments [5].

The morphology studies the pattern of the word and how to frame a word from another word regardless of its position in the sentence, i.e. morphology teaches us how to make the different patterns of derivatives. It concentrate just the variable words, which means the words that can be transformed from a pattern to another of a similar root to express different meanings related to its root.

- Roots (الجذور): set of consonants three to be precise that contain the base meaning of the word. From a root we can generated until 30 words.

Table 1-2: Different derivation of the Root: ك ت ب.

Derivative	كاتِبٌ	مَكْتُوبٌ	مَكْتَبٌ	مَكْتَبَةٌ	كُتِبَ
Meaning	Writer	Written	Desk – office	Library - stationary	Was wrote

- Patterns (الأوزان): set molds of words that roots can be inserted into. Together, the root letters placed inside the patterns are words. Patterns also carry meanings, similar to how suffixes and prefixes do so. The root ف-ع-ل, meaning "to do," is often used to model patterns. In these contexts, ف-ع-ل represents where you will insert the letters of any root into the pattern to create the full meaning.

Table 1-3: Different pattern of the word: كتب ***(write).***

Morphologic pattern	فاعلٌ	مَفْعُولٌ	مَفْعَلٌ	ممَفْعَلَةٌ	فُعِلَ

- Stems: It is the derivation obtained from a given root according to a specific pattern. For example, the stem « مكتـــــــــــــــب, desk», it is obtained from the root «كتب, write» according to the pattern « مفعل».
- Affixes: bound morpheme that is joined before or after the Arabic words, with it we can form new stems by the same root. There is two types of affixes: suffixes (placed after the root) and prefixes (placed before the root).

3.3. Arabic Syntactic

Syntax which is a branch of linguistics is dealing with the internal structure of sentences and the interrelationships among the internal parts. It focuses on grammar that explains the way in which words are arranged to show relationships of meaning within (and sometimes between) sentences and how words put together [5].

The syntax studies the formation of the sentences and the structures, i.e. it teaches us the changes which happen to the words because of its position in the structure. It also studies the signs of the final letters of the words, i.e. it studies the changes which happen because of the position in the sentence or because of the effect of each word towards another.

Enclitic suffix Stem Prefix Proclitic

- Proclitic: Prepositions or Conjunctions.
- Stem: The basis of the word.
- Prefix and Suffix: Grammatical features such as Verb mode, Gender, Number...
- Enclitic: Personal Pronoun.

The following figure 1-1 shows the richness of the Arabic Morphology:

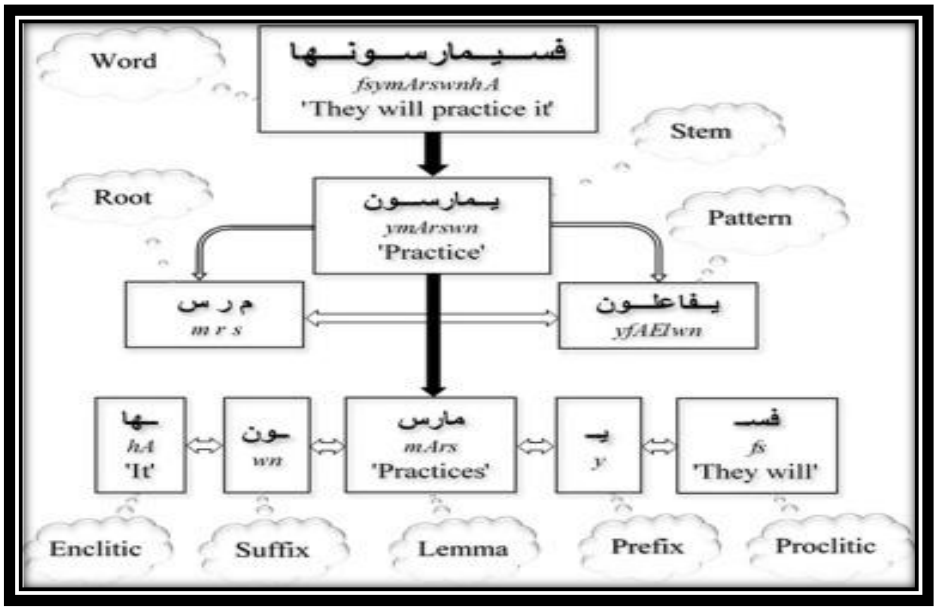


Figure 1-1: Segmentation of an Arabic written word.

3.4. Word Categories

In Arabic a meaningful word is divided into three self-contained categories namely Noun اَلاسْمُ, Verb اَلْفِعْلُ and Particle اَلْحَرْفُ [1] .

- Verb or "Fi'l" اَلْفِعْلُ: denotes an action, in either: Past, Imperfect Tenses: Present, Future. In Modern Standard Arabic, verb conjugations are marked by:

  - Person: who is committing an action (e.g. I, she, you, etc.);
  - Number: singular (for one person or thing), dual (for exactly two), or plural (for many);
  - Tense: Arabic has present المضارع (e.g. 'أفعل, I do') and past الماضي (e.g. 'فعلت, I did') tenses;
  - Gender: masculine or feminine;
  - Mood: the speaker's attitude to the action, e.g. indicative ('I go'), subjunctive ('so that you go').

- Noun or "Ism" اَلاسْمُ: A word that names a person, place, or thing. It can be categorized into two categories with respect to its gender: Masculine اَلْمُذَكَّرُ and Feminine اَلْمُؤَنَّثُ. And into three categories with respect to number: Singular اَلْمُفْرَدُ, Dual اَلْمُثَنَّى and Plural اَلْجَمْعُ. The declination of names is done according to those guidelines:

  - Feminine singular: we add ة, for example "ممثل, actor" become" ممثلة, actress".
  - Feminine Dual: we add ين or ون according to a word position in a sentence and the ة become ت, for example "ممثلة, actress" become "ممثلتان, actresses" or "ممثلتين, actresses".
  - Feminine Plural: we add ات, for example "ممثل, actor" become" ممثلات, actresses".

[1] https://www.learningarabicwithangela.com/post/understanding-the-arabic-word.

- Masculine Dual: we add ين or ان, for example "ممثل, actor" become "ممثلان" or "ممثلين, actors"
- Masculine Plural: we add ين or ون according to a word position in a sentence, for example "ممثل, actor" become "ممثلين, actors" or "ممثلون, actors".

- Particle or "Harf" اَلْحَرْفُ: does not have a meaning on its own and is comprehended from the setting of the sentence and words in it such as articles, conjunctions, prepositions and others.

For example, the following figure 1-2 summarizes what have been said in this section.

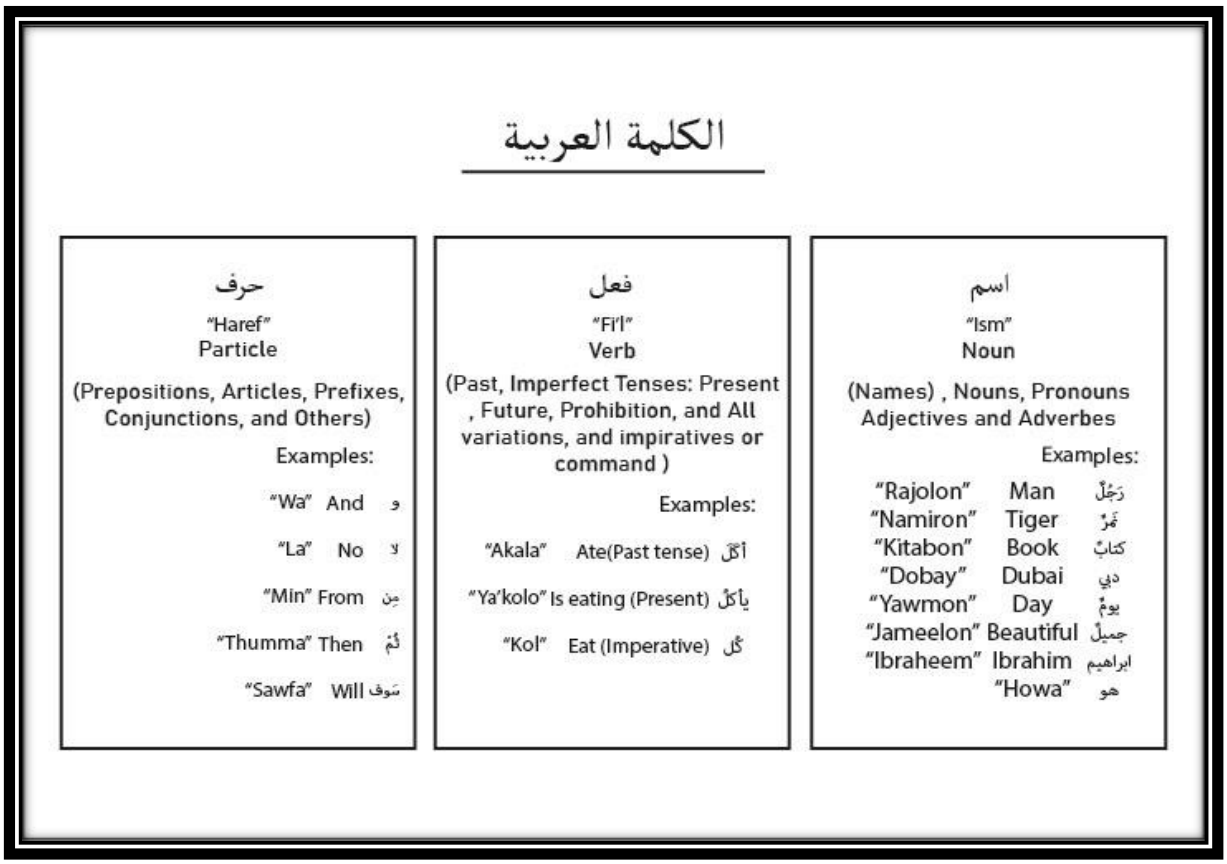


Figure 4-2: The Arabic Word: Noun; Verb; Particle; with examples. [1]

[1] https://www.learningarabicwithangela.com/post/understanding-the-arabic-word.

4. Arabic Natural Language Processing Difficulties

Among the problems of the Arabic language we quote ambiguity, absence of vowels, segmentation and agglutination.

4.1. Ambiguity

The Arabic words can be ambiguous at Lexical and Grammatical level.

Examples:

The word "مغرب" it is lexically ambiguous; it can be Morocco (The country) or prayer time (أذان المغرب)

The word "علم" it is grammatically ambiguous; it can belong to two different grammatical categories noun and verb. Verb 'alama as teaches and noun 'alam as flag.

There is also syntactic ambiguity, the same sentence can have several possible meanings depending on its syntactic interpretations.

4.2. Absence of vowels

As a general rule, only the Coran and texts of a didactic nature are vocalized; short vowels, repetition and lengthening, carried by diacritical signs, do not appear in everyday texts. This characteristic leads to a high degree of ambiguity, since these non-vocalized forms are nowadays found very frequently, and this ambiguity can usually be removed by associating the form with meaning and context, etc. For example, the effect of the non-vowel word like شعر generates ambiguity between شعر/poetry, شعر/hair and شعر/sense. This ambiguity could, in some cases, be removed either by a deep syntactic and semantic analysis of the sentence, or by a statistical analysis as in the case of search engines.

4.3. Agglutination

Unlike Latin languages, Arabic is an agglutinating language; articles, prepositions and pronouns stick to the adjectives, nouns, verbs and particles to which they refer, i.e. the set of morphemes stuck together and constituting a lexical unit convey several morpho-syntactic information which creates morphological ambiguity during word analysis [4].

Figure 1-3: Agglutination example «وبقوله».

4.4. Segmentation

The segmentation of text written in any Arabic script is a most difficult task. To analyses a text, we need to segment it into paragraphs, sentences and words. The main problem lies in the fact that this segmentation is a source of ambiguity, since, on the one hand, punctuation is rarely used in Arabic texts, and on the other hand, this punctuation, when it exists, is not always decisive in guiding segmentation. In addition, some tool words may mark the beginning of a new sentence, which requires surface analysis in order to segment the text. Sentences (1) and (2) illustrate this problem:

"تعلم الطفل القراءة والكتابة في المدرسة/ the child has learned to read and write at school" (1). In this sentence, the particle 'و, w' does not act as a separator between propositions but as a coordinating conjunction between the words "القراءة" and "الكتابة", and therefore does not segment the sentence.

“تعلم الطفل القراءة في المدرسةو حفظ القرآن الكريم في المسجد / the child learned to read at school, and learned the Coran at the mosque" (2).

While the particle "و, w" in (2) acts as a separator between propositions and segments the sentence into two propositions. In general, space remains the main criterion for text segmentation [4].

5. Arabic Natural Language Processing Works

The researchers targeted the Arabic language in its three main varieties: Classical Arabic (CA), Modern Standard Arabic (MSA) and Arabic Dialect (AD) [6].

5.1. Works on Classical Arabic

In this section, we will present some works in Classical Arabic (CA).

5.1.1. Basic Language Analyses

Dukes et al. presented an annotated linguistic resource (A part of Quranic Arabic Corpus noted QAC) the corpus provides three levels of analysis: morphological annotation, a syntactic treebank and a semantic ontology. These authors introduced traditional Arabic grammar and describing the annotation process, including the syntactic relations used to label dependency graphs. They also highlight key parts of the full annotation guidelines such as: Verbs, Subjects and Objects and pronoun.

5.1.2. Building Resources

In 2012, Sharaf and Atwell presented a manually annotated large corpus “QurSim”, created from the original Quranic text, where semantically similar or related verses are linked together. They also presents in the same year “QurAna”, a large corpus created from the same source, where personal pronouns are tagged with their antecedents. In 2016, Belinkov et al. proposed a large-scale historical Arabic corpus. These authors lemmatized the entire corpus in order to use it in semantic analysis. Zerrouki and Balla in 2017, proposed a large freely available vocalized

corpus, containing 75 million words, collected from freely published texts in old books.

5.1.3. Language Identification

In 2016, Asda et al. proposed the development of Quran reciter recognition and identification system, based on Mel-Frequency Cepstral Coefficient (MFCC) feature extraction and Artificial Neural Networks. From every speech, characteristics from the utterances will be extracted through neural network models. The proposed system is divided into two parts. The first part consists of feature extraction and the second part is the identification process using neural networks.

5.1.4. Semantic-Level Analysis

The team of Eric Atwell at Leeds University worked on the Quran analysis project, in order to build a Semantic Search and Intelligence System. This system provides manual users with the ability to search the Quran semantically and analyze all aspects of the text. Another project dealing with a different task including translation is the Tanzil[1] project. Among the applications of Tanzil features we cite the translation of Quran into other languages such as: English, German, Italian, etc.

5.2. Works on Modern Standard Arabic

In this section, we will present some works in Modern Standard Arabic (MSA).

[1] Tanzil is a Quranic project launched in early 2007 to produce a highly compliant unicode Quran text used in Quranic websites and applications.

5.2.1. Basic Language Analyses

Abdelali et al. presented a fast and accurate Arabic segmenter called "Farasa"[1]. This approach is based on SVM using linear kernels. To validate Farasa, the authors compare it with two other segmenters: MADAMIRA[2] and the Stanford Arabic Segmenter (SAS). Khalifa et al. presented YAMAMA, a morphological analyzer, focused on MSA and EGY dialect. It was inspired by the fast execution of Farasa and the rich output of MADAMIRA. More recently, Zalmout and Habash (2017) introduced a model for Arabic morphological disambiguation based on Recurrent Neural Networks (RNN). The authors based their work on Penn Arabic Treebank (PATB) and they used Long Short-Term Memory model (LSTM) by showing that this model provides a significant performance.

5.2.2. Building Resources

In 2015, Selab and Guessoum constructed the TALAA corpus, a large Arabic corpus containing 14 million words, built from daily Arabic newspaper websites. In the same year, Yousfi et al. presented the dataset ALIF dedicated to Arabic embedded text recognition in TV broadcasts. It is composed of a large number of manually annotated text images that were extracted from Arabic TV broadcasts.

5.2.3. Identification and recognition

Many works are oriented on Arabic identification (Ali et al., 2016, El Haj et al., 2017, Shon et al., 2018, Tachicart et al., 2017). They dealt with the Multi dialectal identification as well as the MSA one where the last one proposes an identification system distinguishing between the Moroccan

[1] Farasa (which means "insight" in Arabic), is a fast and accurate text processing toolkit for Arabic text.
[2] MADAMIRA is a Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic.

dialect and MSA. However, as the main focus of these works is related to dialect identification.

5.2.4. Semantic-level analysis and synthesis

Almost all of the MT system are based on statistical machine translation (using Moses toolkit) which requires a parallel corpus. In this context, Inoue et al. (2018) presented the first results of Arabic-Japanese phrase-based MT by relying on the alignment of 900 documents using two techniques: manual and automatic.

5.3. Works on Arabic Dialects

In this section, we will present some works in Arabic Dialects (AD).

5.3.1. Basic Language Analyses

Many works have been proposed in order to offer a set of Orthographic rules, standards and conventions. The work presented by Saadane and Habash follows the previous efforts made and demonstrated for Egyptian and Tunisian dialects and applies them for the Algerian0 dialect. The purpose of Habash et al. was to present a common set of guidelines with enough specificity to help in creating dialect specific conventions. Segmentation and Part-of-speech (POS) tagging are two of the most important addressed areas NLP. Salloum and Habash presented ADAM (Analyzer for Dialectal Arabic Morphology). The authors evaluated ADAM's performance on LEV and EGY. Zribi et al. proposed a method adapting a MSA morphological analyzer for the Tunisian dialect (TD).

5.3.2. Building Resources

Kwaik et al. presented the construction of the Shami corpus, a LEV Dialect Corpus. This corpus covers data from the four dialects spoken in Palestine, Jordan, Lebanon and Syria and contains 117,805 sentences. Jarrar

et al. presented the construction of *Curras*, a morphologically annotated corpus of the Palestinian Arabic dialect. *Curras* consists of more than 56,000 tokens, which were annotated with rich morphological and lexical features. Al-Twairesh et al. proposed SUAR, a semi-automatically Saudi corpus which was morphologically annotated automatically using the MADAMIRA. The generated corpus was manually checked the resulted corpus contains 104,079 words.

5.3.3. Language Identification

Ali et al., used different approaches for dialect identification in Arabic broadcast speech focused on multi dialects (EGY, LEV, GLF, and MAGH as well as MSA). Their methods are based on phonetic and lexical features obtained from a speech recognition system. El Haj et al. presented an approach of Arabic dialect identification using language bivalency[1] and written code-switching. The authors concentrate on multi dialects as well as MSA. For the classification task, the authors use different algorithms: NB, SVM, etc. More recently Ali (2018) proposed a character-level convolution neural network model for distinguishing between MSA and multi dialects.

5.3.4. Semantic-level analysis

In the context of MT, Meftouh et al. used a phrase-based MT system, GIZA++[2] for alignment and SRILM[3] toolkit. The best results that these authors obtained were between the Algiers dialect and the dialect of Annaba

[1] Bivalency is word or element is treated by language users as having a fundamentally similar semantic content in more than one language or dialect.
[2] Giza++ is the automatic word alignment tool for statistical machine translation. Word alignment is the task of identifying translational relations between words in parallel corpora with the aim of re-using them in natural language processing.
[3] SRILM is a toolkit for building and applying statistical language models (LMs), primarily for use in speech recognition, statistical tagging and segmentation, and machine translation.

with BLEU score up to 67.31 which is perfectly understandable where both dialects are spoken into the same country (Algeria).

In the context of SA, El-Beltagy implemented a simple sentiment analysis task using the bag of words model, with uni-gram and bi-gram TF-IDF weights. As a classifier, the authors used NB in combination with the constructed lexicon NileULex[1]. The results show that the integration of NileULex improved the results of classification (F1-score up to 79%). Guellil et al. propose a sentiment analysis algorithm dealing with Algerian dialect morphology and handling negation and opposition, the best F1-score that they achieved is up to 78%. To evaluate the performance of their corpus, Medhaffar et al. relied on three classifiers such as SVM and NB and MultiLayer Perceptron classifier (MLP).

The following table shows the comparison between types of works in Arabic Natural Language Processing.

Table 1-4: Comparison between ANLP works.

Type	Field	Authors/Year	Description
Classical Arabic	Basic Language Analyses	Dukes et al (2010)	An annotated linguistic resource, provides three levels of analysis: morphological annotation, a syntactic treebank and a semantic ontology.

[1] NileULex is an Arabic sentiment lexicon containing close to six thousands Arabic words and compound phrases. Forty five percent of the terms and expressions in the lexicon are Egyptian or colloquial while fifty five percent are Modern Standard Arabic.

	Building Resources	Sharaf and Atwell (2012)	Two large corpus "QurSim" and "QurAna", created from the original Quranic text.
		Belinkov et al (2016)	Large-scale historical Arabic corpus.
		Zerrouki and Balla (2017)	Large freely available vocalized corpus, containing 75 million words.
	Language Identification	Asda et al (2016)	Development of Quran reciter recognition and identification system, based on MFCC feature extraction and Artificial Neural Networks.
	Semantic-Level Analysis	The team of Eric Atwell at Leeds University	Quran analysis project, a system provides manual users with the ability to search the Quran semantically and

			analyze all aspects of the text. The Tanzil project dealing with a different task including translation (2007).
Modern Standard Arabic	Basic Language Analyses	Abdelali et al (2016)	Fast and accurate Arabic segmenter called “Farasa”. The authors compare it with two other segmenters (MADAMIRA and the Stanford Arabic Segmenter (SAS)).
		Khalifa et al (2016)	YAMAMA a morphological analyzer, focused on MSA and EGY dialect.
		Zalmout and Habash (2017)	A model for Arabic morphological disambiguation based on Recurrent Neural Networks (RNN).
	Building Resources	Selab and Guessoum (2015)	The TALAA corpus, a large Arabic corpus containing 14

			million words, built from daily Arabic newspaper websites.
		Yousfi et al (2015)	ALIF dataset dedicated to Arabic embedded text recognition in TV broadcasts.
	Language Identification	Ali et al (2016) El Haj et al (2017) Shon et al (2018) Tachicart et al (2017)	They dealt with the Multi dialectal identification as well as the MSA one where the last one proposes an identification system distinguishing between the Moroccan dialect and MSA.
	Semantic-Level Analysis	Inoue et al (2018)	Presented the first results of Arabic-Japanese phrase-based MT by relying on the alignment of 900 documents using two techniques: manual and automatic.

Arabic Dialects	Basic Language Analyses	Habash et al (2018)	A common set of guidelines with enough specificity to help in creating dialect specific conventions.
		Salloum and Habash (2014)	Presented ADAM (Analyzer for Dialectal Arabic Morphology).
		Zribi et al (2013)	Proposed a method adapting a MSA morphological analyzer for the Tunisian dialect (TD).
	Building Resources	Kwaik et al (2018)	Present the construction of the Shami corpus which covers data from the four dialects spoken in Palestine, Jordan, Lebanon and Syria and contains 117,805 sentences.
		Jarrar et al (2017)	*Curras*, a morphologically annotated corpus of the Palestinian Arabic

			dialect. *Curras* consists of more than 56,000 tokens.
		Al-Twairesh et al (2018)	Proposed SUAR, a semiautomatically Saudi corpus which was morphologically annotated automatically using the MADAMIRA, the resulted corpus contains 104,079 words.
	Language Identification	Ali et al (2016)	Used different approaches for dialect identification in Arabic broadcast speech focused on multi dialects (EGY, LEV, GLF, and MAGH as well as MSA).
		El Haj et al (2017)	Presented an approach of Arabic dialect identification using language "bivalency" and written code-switching.
	Semantic-Level Analysis	Meftouh et al (2018)	Used a phrasebased MT system, GIZA for alignment and SRILM toolkit The best results that these authors obtained were between

			the Algiers dialect and the dialect of Annaba with BLEU score up to 67.31.
		El-Beltagy (2016)	Implemented a simple sentiment analysis task (using BOW, uni-grams , bi-grames and TF IDF weights, NB classifier) The results show that the integration of NileULex improved the results of classification (F1-score up to 79%).

Besides, there are some other works in the Arabic poetry that will be introduced in chapter number three.

6. Conclusion

The aim of this chapter was to present the Arabic Natural Language Processing filed, its challenges and objectives. Then we introduced the Arabic language including its particularities, morphology, syntactic and word categories.

Finally, we talked about the different difficulties of the Arabic Natural Language Processing such as Ambiguity, Agglutination, and Absence of vowels and Segmentation. Then, we moved to the ANLP works.

In the next chapter, we will discuss the text classification, its processing, types, and the Arabic Classification and its difficulties.

Chapter II:
Text Classification

1. Introduction

The Internet is a very effective technique for obtaining a huge amount of information in different forms such as documents. Mainly, there are millions of documents from various sources, most of which contain valuable information. Manual classification of documents consumes time and is very difficult, especially when people must estimate the category based on the information included. Therefore, the automatic text classification is used to discover the basic information of text documents automatically while saving human effort and time [7].

The automated classification of texts into predefined categories has witnessed a booming interest in the last 10 years, due to the increased availability of documents in digital form and the ensuing need to organize them and that is because of the large amount of information exists in text documents (over 80% is stored as text). Therefore, it is important to use text mining (or text classification) to discover knowledge from these unstructured data. Automatic text classification considered as one of important applications in text mining [8].

2. Text Classification

Text classification is a classic topic for natural language processing, in which one needs to assign predefined categories to text documents. It is the process of assignment of unclassified text to appropriate classes based on their content [8].

Automated text classification has been considered as a vital method to manage and process a vast number of documents in digital forms that are widespread and continuously increasing. In general, text classification plays an important role in Information Extraction and Summarization, Text Retrieval, and Question Answering [9].

Text categorization is the task of assigning a Boolean value to each pair (dj, ci) $\in$ D×C, where D is a domain of documents and C= {c1, ... , ci} is a set of predefined

categories. A value of T (True) assigned to (dj,ci) indicates a decision to file dj under ci, while a value of F (False) indicates a decision not to file dj under ci. More formally, the task is to approximate the unknown target function Φ: D×C→ {T, F} (that describes how documents ought to be classified) by means of a function Φ: D×C → {T, F} called the classifier (aka rule, or hypothesis, or model) such that Φ and Φ "coincide as much as possible" [10]. The figure 2-1 bellow elaborate text classification.

***Figure 2-1: Text Classification** [11].*

2.1. Text Classification Process

After a several of reading about text classification process, we came up with these five phases: pre-processing, document representation, dimension reduction, classification and finally evaluation step.

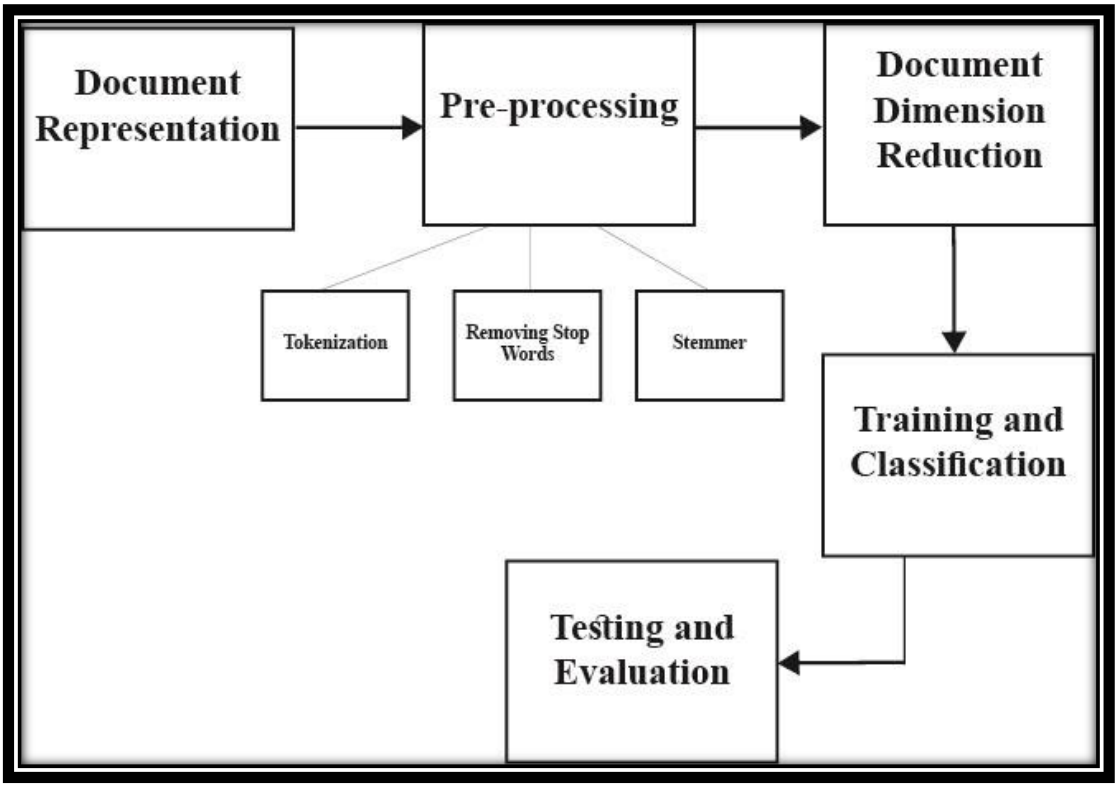


Figure 2-2: Text Classification Process.

2.1.1. Preprocessing

Most of the text and documents contain many words that are redundant for text classification, such as stop words. In this section, we briefly explain some techniques and methods for text cleaning and pre-processing text documents. In many algorithms like statistical and probabilistic learning methods, noise and unnecessary features can negatively affect the overall performance. So, the elimination of these features is extremely important.

2.1.1.1. Tokenization

Tokenization is commonly understood as the first step of any kind of natural language text preparation. The major goal of this task is to convert a stream of characters into a stream of processing units called tokens [12].

Example:

Input: “People are dying because of the Corona Virus.”

Output: {‘People’, ‘are’,’dying’,’because’,’of’,’the’,’Corona’,’Virus’,’.’}

2.1.1.2. Removing Stop Words

Stop words are the extremely common and semantically non-selective words. Punctuations and numbers, if deemed irrelevant to the classification task at hand are removed, although in some cases these may be informative and thus retained. Stop words are known to have low information content such as conjunctions and prepositions that is why they need to be deleted.

Table 2-1: Stop Words.

Arabic	English
متى	When
في	In
أنا	Me

Example:

This example shows us how a sentence become after removing stop-words

Input: “People are dying because of the Corona Virus.”

Output: “People dying Corona Virus.”

2.1.1.3. Stemming

Stemming is another common preprocessing step. In order to reduce the size of the initial feature set is to remove misspelled or words with the same stem. A stemmer (an algorithm which performs stemming), removes words with the same stem and keeps the stem or the most common of them as feature [9].

Example: ‘train’, ‘training’, ‘trainer’ , ’trains’ → ‘train’

2.1.2. Document representation

The representation of texts is a very important step in the process of Text Classification, where terms are words, phrases, or any other indexing units used to recognize the contents of a text, so it is necessary to use an effective representation technique allowing the texts to be represented in a machine-usable form. The different methods that exist for the representation of texts are [13]:

2.1.2.1. Bag of words Representation

This representation of the texts is the simplest. It was introduced as part of the vector model. Texts are simply transformed into vectors, each component of which represents a term.

2.1.2.2. Representation of texts with Lemmas

Lemmatization consists in using grammatical analysis to replace verbs by their infinitive form and nouns by their singular form. Lemmatization is therefore more complicated to implement than the search for roots, since it requires a grammatical analysis of the texts.

2.1.2.3. Representation of texts with Lexical Roots

This method consists in replacing the words of the document by their lexical roots, and in grouping the words of the same root in a single component. Thus, several words of the document will be replaced by the same root, several algorithms have been proposed. We can cite the Porter algorithm and the Khodja algorithm for the Arabic language.

Indeed, a root can be common for words with different meanings such as the words genre, gender, generic have the same root "genus" but refer to three different notions, and this representation also depends on the language used.

2.1.2.4. Representation of texts with N-grams

This method consists of representing the document by n-grams. The n-gram is a sequence of n consecutive characters. It consists of cutting the text into several sequences of n characters while moving with a window of a character. An n-gram of size 1 is called a uni-gram, of size 2 is a bi-gram and size 3 is a tri-gram. This technique has several advantages. The n-grams automatically capture the roots of the most frequent words without going through the lexical roots search step. The lexical roots are language-independent and spaces are taken into account.

2.1.2.5. Representation of texts by Sentences

A certain number of researchers propose to use the sentences as a unit of representation instead of the words as is the case in the representation "bag of words", since the sentences are more informative than the words alone, for example: "retrieval information "," world wide web ", have a lower degree of ambiguity than the constituent words, but also because the sentences have the advantage of preserving the information relating to the position of the word in the sentence.

2.1.3. Documents Dimension Reduction

While stemming and other lemmatization techniques could lessen the number of words, it still presents an issue when using Machine Learning techniques for text Classification because of thousands of words in a document, so it is not possible to do the classification for all those words as features. Also, the computer could have problems processing such amount of data. That is why it is important to select the most representative features as inputs for the classification step and reducing the dimensionality of a document [14].

Alternative methods are suggested to reduce the dimensionality of the set of features. This includes ranking the ability of a term concerning whether it is a good indicator for documents as well as classes. In this way, useful information for discarding terms can be provided by using measures such as Chi square, mutual information, information gain, or term frequency...etc [15]. Figure 3.2 comprise six

different term weighing methods based on supervised and unsupervised methods for Feature Selection (or Feature Elimination):

Methods	Term weighing factors	Denoted by	Description
Supervised term weighing methods	Chi square	χ^2	Multiply *tf* by χ^2 funtion
	Information gain	*ig*	Multiply *tf* by *ig* funtion
	Odd ratio	*OR*	Multiply *tf* by OR funtion
	Relevance factor	*rf*	Multiply *tf* by *rf* funtion
Unsupervised term weighing methods	Term frequency	*Tf*	Number of times term occur in adocuments
	Inverse document frequency	*Idf*	Multiply *tf* by idf funtion

Figure 2-3: Different term weighing methods [16].

The focus of Document Dimension reducing or Feature Selection is to select a subset of variables from the input which can efficiently describe the input data while reducing effects from noise or irrelevant variables and still provide good prediction results. Feature elimination or Feature Selection does not create new features since it uses the input features itself to reduce their number. Each term in a document vector must be associated with a value called weight, which measures the importance of this term and denotes how much this term contributes to the categorization task of the document [16].

There is another method for Dimension Reduction which is Feature Transformation, it is to create new features using the existing features.

2.1.4. Classification Algorithms

There are several methods used to classify text such as Support Vector Machine (SVM), K Nearest Neighbor (KNN), Artificial Neural Networks (ANN), Naive Bayes Classifier, and Decision Trees.

2.1.4.1. K- Nearest Neighbor

To classify an unknown document vector d, the k-nearest neighbor (k-NN) algorithm ranks the document's neighbors among the training document vectors, and use the class labels of the k most similar neighbors to predict the class of the input document. The classes of these neighbors are weighted using the similarity of each neighbor to d, where similarity may be measured by for example the Euclidean distance or the cosine between the two document vectors.

k-NN has been applied to text categorization since the early days of its research. However, it has a set of drawbacks. k-NN is a lazy learning example-based method that does not have an off-line training phase [17]. Some of the advantage and disadvantage of k-NN can be summed up in table 2-2.

Table 2.1-2: Advantages and Disadvantages of KNN.

Advantages	Disadvantages
- Very simple to implement and understand; - Highly effective for many classification problems, especially with low dimensionality.	- Hard to find out the number of k, time cost is more; - Not suitable for high dimensionality problems; - Computationally intensive, especially with a large training set.

2.1.4.2. Naïve Bayes

The Naive Bayes (NB) classifier is a probabilistic model that uses the joint probabilities of terms and categories to estimate the probabilities of categories given a

test document. The naive part of the classifier comes from the simplifying assumption that all terms are conditionally independent of each other given a category. Because of this independence assumption, the parameters for each term can be learned separately and this simplifies and speeds the computation operations compared to non- naive Bayes classifiers [17]. Some of the advantage and disadvantage of Naïve Bayes can be summed up in table 2-3.

Table 2-3: Advantages and Disadvantages of Naive Bayes.

Advantages	Disadvantages
- Easy for implementation and computation; - Surprisingly accurate for a large set of problems, scalable to very large data sets, and is used for many NLP models.	- Very poor when feature are co-related to each other; - Problems where categories may be overlapping or there are unknown categories can dramatically reduce accuracy.

2.1.4.3. Support Vector Machine

Support Vector Machines (SVM) is just one out of many algorithms we can choose from when doing text classification. Like naive bayes, SVM does not need much training data to start providing accurate results. Although it needs more computational resources than Naive Bayes, SVM can achieve more accurate results.

In short, SVM takes care of drawing a “line” or hyperplane that divides a space into two subspaces: one subspace that contains vectors that belong to a group and another subspace that contains vectors that do not belong to that group. Those vectors are representations of your training texts and a group is a tag you have tagged your texts with [18]. Some of the advantage and disadvantage of SVM can be summed up in table 2-4.

Table 2-4: Advantage and Disadvantage of SVM.

Advantage	Disadvantage
- Compact of description of the learned model, more capable to solve multi-label classification.	- Training speed is low.

2.1.4.4. Neural Network

A neural network classifier is a network of units, where the input units usually represent terms, the output units represents the category. For classifying a test document, its term weights are assigned to the input units; the activation of these units is propagated forward through the network, and the value that the output units takes up as a consequence determines the categorization decision [8]. Some of the advantage and disadvantage of NN can be summed up in table 2-5.

Table 2-5: Advantages and Disadvantages of NN.

Advantages	Disadvantages
- Provide better result in complex domain non-exhaustive category sets and complex functions relating input to output variables; - Powerful tuning options to prevent over- and under-fitting.	- Long training process; - Theoretically complex, difficult to implement; - Non-intuitive and requires expertise to tune.

2.1.4.5. Decision Tree

Decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called “root” that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves. Such a classifier categorizes a test document d_j by recursively testing for the weights that the terms labeling the internal

nodes have in vector $\overrightarrow{d_j}$, until a leaf node is reached; the label of this node is then assigned to d_j [10]. Some of the advantage and disadvantage of Decision Tree can be summed up in table 2-6.

Table 2-6: Advantage and Disadvantage of Decision Tree.

Advantage	Disadvantage
- Simple even non expert user can understand; - Able to model complex decision processes, very intuitive interpretation of results.	- Irrelevant attributes may affect badly the construction of a decision tree; - Can very easily over fit the data.

The following table 2-7 present the above represented algorithms with their formula.

Table 2-7: Different classification algorithms [16].

Classification algorithm	Formula	Classifier Type
KNN	Distance measured by Euclidian distance $dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$	Multiclass
NB	Posterior probability $P(H\|X) = \frac{P(X\|H)P(H)}{P(X)}$	Multiclass
SVM	$minmax = \left\{\frac{1}{2} \parallel w \parallel^2 - \sum_{i}^{n} \alpha_i [y_i(w\_x_i - b) - 1]\right\}$	Multiclass

NN	$I_j = \sum_j w_{ij} O_i + \theta_j$ J is hidden layer net input $O_J = \frac{1}{1 + e^{-1_j}}$ Output unit	Either binary or multiclass
Decision Tree	Partition of data, which is a set of training tuples and their associated class labels, then by making set of candidate attributes select the attribute by attribute selection methods.	Multiclass

2.1.5. Evaluation Measure

To evaluate performance of text classifier first calculates precision and recall. There are various methods to determine effectiveness; however, precision, recall, and accuracy are most often used. To determine these, one must first begin by understanding if the classification of a document was a true positive (TP), false positive (FP), true negative (TN), or false negative (FN).

- Recall (*R*): is defined as the probability that, if it should be classified under random document *dx* category (*ci*), this decision is taken [9].

$$R = \frac{TP}{TP + FN}$$

- Precision (P): is determined as the conditional probability that a random document *d* is classified under (*ci)*, or what would be deemed the correct category. It represents the classifiers ability to place a document as being under the correct category as opposed to all documents place in that category, both correct and incorrect [9].

$$P = \frac{VP}{VP + FP}$$

- TP (True Positive): determined as a document being classified correctly as relating to a category.
- FP: (False Positive): determined as a document that is said to be related to the category incorrectly.
- FN (False Negative): determined as document as a document that is not marked as related to a category but should be.
- TN (True Negative): determined that should not be marked as being in a particular category and are not.

- Accuracy: is commonly used as a measure for categorization techniques. Accuracy values, however, are much less reluctant to variations in the number of correct decisions than precision and recall [9].

$$A = \frac{(VP + TN)}{(VP + VN + FP + FN)}$$

- F-score: also called the F1 score or F measure, is a measure of a test's accuracy. The F score is defined as the weighted harmonic mean of the test's precision and recall.

$$F = 2\ \frac{P \times R}{T + R}$$

- Kappa (K): the Kappa coefficient used to verify the presence of the themes that were presented and it is also a statistical measure; it is varies from 0 to 1, where:

Table 2-8: Kappa Accord.

Cohen's Kappa statistic	Strength of agreement
< 0.00	Poor
0.00 – 0.20	Slight
0.20 – 0.40	Fair
0.41 – 0.60	Moderate
0.61 – 0.80	Substantial
0.81 – 1.00	Almost perfect

There are other evaluation measures and techniques such as complexity and time.

2.2. Text Categorization Types

Automatic classification consists of grouping various objects into subsets of objects (classes). It can be supervised where the classes are known a priori, they generally have an associated semantics or unsupervised (Clustering) where the classes are based on the structure of the objects, the semantics associated with the classes is more difficult to determine.

2.2.1. Supervised Classification

Supervised classification is one of the machine learning approaches. It relies on an analyst to define the classes that the data are classified into and provide the training data of each defined class. The output of the trained classifier with the training data is an assignment of a class label to each input data point. Several classifiers are widely used, including a minimum distance classifier, maximum likelihood classification, K-nearest neighbors, and support vector machines.

- **Process**

Supervised classification of text is done when you have defined the classification categories. It works on training and testing principle. We feed labeled data to the machine learning algorithm to work on. The algorithm is trained on the labeled dataset and gives the desired output (the pre-defined categories). During the testing phase, the algorithm is fed with unobserved data and classifies them into categories based on the training phase [11].

2.2.2. Unsupervised Classification

Unsupervised classification is also known as data clustering and is defined as the problem of finding homogeneous groups of data points in a given multidimensional data set. Each of these groups is called a cluster and defined as a region in which the density of objects is locally higher than in other regions. One objective of unsupervised learning is construction of decision boundaries based on unlabeled training data. Typical clustering algorithms depend on a choice of a similarity measure between data points, and a "correct" clustering result depends on an appropriate choice of a similarity measure.

- **Process**

Unsupervised classification is done without providing external information. Here the algorithms try to discover natural structure in data. Please note that natural structure might not be exactly what humans think of as logical division. The algorithm looks for similar patterns and structures in the data points and groups them into clusters. The classification of the data is done based on the clusters formed. Take web search for an example. The algorithm makes clusters based on the search term and presents them as results to the user [11].

The difference of classification and clustering algorithms shows in figure 2-5.

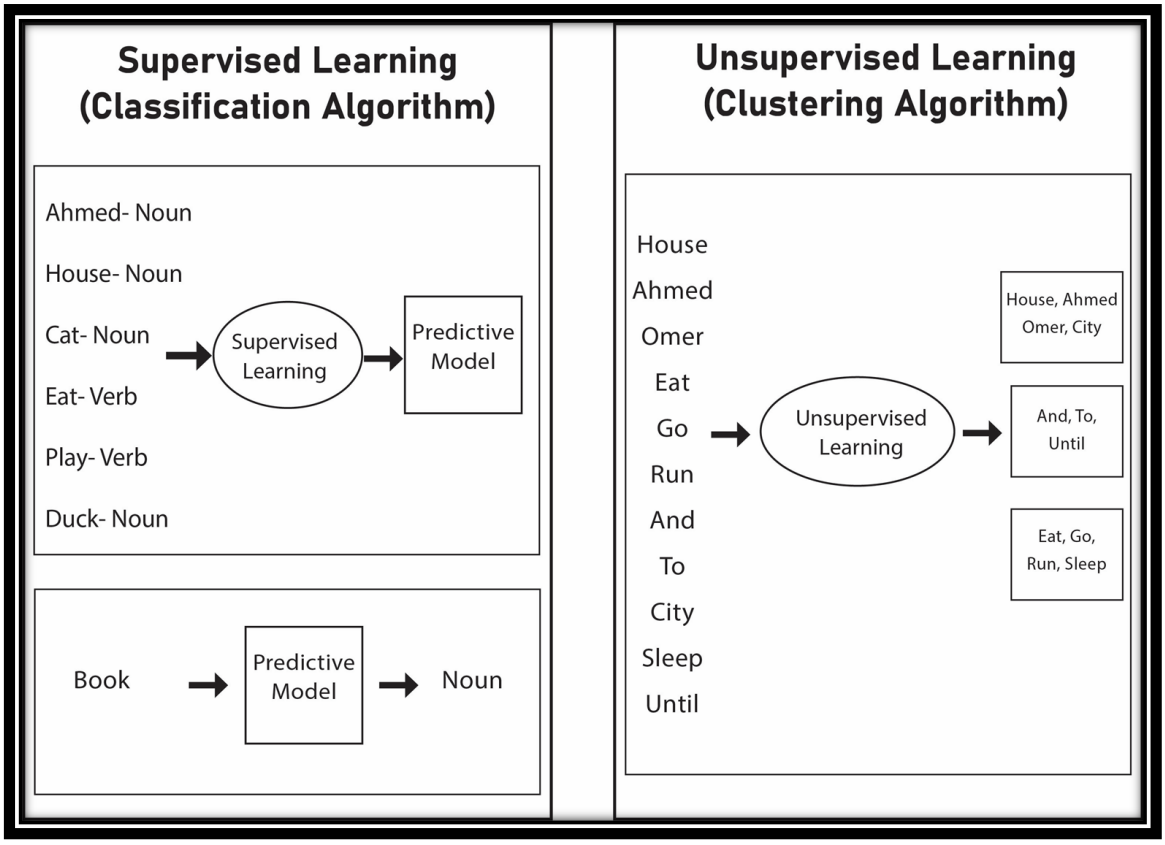


Figure 2-5: Example shows how classification and clustering algorithms work.

The figure 2-5 above explained that in classification algorithm we have to train our data to create a predictive model that we will use to predict the class of the test data which means that we will give it a new data that it did not see it while training. Unlike in clustering algorithm when we gave it data; it classify them according to their classes without using a predictive model.

The following table is a list of points, describe the comparisons Between Supervised Learning and Unsupervised Learning:

Table 2-9: Supervised Classification Vs Unsupervised Classification.

Basis	Supervised Classification	Unsupervised Classification
Method	- Input variables and output variables will be given.	- Only input data will be given.
Goal	- Supervised learning goal is to determine the function so well that when new input data set given, can predict the output; - Supervised classification is basically asking computers to imitate humans.	- The unsupervised learning goal is to model the hidden patterns or underlying structure in the given input data in order to learn about the data.
Examples	- Classification; - Regression; - Linear regression; - Support vector machine.	- Clustering; - Association; - k-means; - Association.
Uses	- Supervised learning is often used for export systems in image recognition, speech recognition, forecasting, financial analysis and	- Unsupervised learning algorithms are used to pre-process the data, during exploratory analysis or to pre-train supervised learning algorithms.

	training neural networks and decision trees etc.	

2.3. Text Classification Applications

Text Classification has been used for a number of different applications, of which we here briefly review some of them [10]:

2.3.1. Automatic Indexing for Boolean Information Retrieval Systems

The application that has spawned most of the early research in the field is that of automatic document indexing for IR systems relying on a controlled dictionary, the most prominent example of which is Boolean systems. Each document is assigned one or more key words or key phrases describing its content, where these key words and key phrases belong to a finite set called controlled dictionary, often consisting of a thematic hierarchical thesaurus.

If the entries in the controlled vocabulary are viewed as categories, text indexing is an instance of Text Classification. Various text classifiers explicitly conceived for document indexing have been described in the literature.

2.3.2. Document Organization

Indexing with a controlled vocabulary is an instance of the general problem of document base organization. In general, many other issues pertaining to document organization and filing, be it for purposes of personal organization or structuring of a corporate document base, may be addressed by TC techniques. For instance, at the offices of a newspaper incoming “classified” ads must be, prior to publication, categorized under categories such as Personals, Cars for Sale, Real Estate, etc.

Newspapers dealing with a high volume of classified ads would benefit from an automatic system that chooses the most suitable category for a given ad.

2.3.3. Word Sense Disambiguation

Word Sense disambiguation is the activity of finding, given the occurrence in a text of an ambiguous (polysemous or homonymous) word, the sense of this particular word occurrence.

Word Sense Disambiguation is very important for many applications, including natural language processing, and indexing documents by word senses rather than by words for Information Retrieval purposes. WSD may be seen as a Text Classification task once we view word occurrence contexts as documents and word senses as categories.

Word Sense Disambiguation is just an example of the more general issue of resolving natural language ambiguities, one of the most important problems in computational linguistics. Other examples, which may all be tackled by means of Text Classification techniques: context-sensitive, spelling correction, prepositional phrase attachment and part of speech tagging...etc.

2.3.4. Text Filtering

Text filtering is the activity of classifying a stream of incoming documents dispatched in an asynchronous way by an information producer to an information consumer. A typical case is a newsfeed, where the producer is a news agency and the consumer are a newspaper.

The explosion in the availability of digital information has boosted the importance of such systems, which are nowadays being used in contexts such as the

creation of personalized Web newspapers, junk e-mail blocking, and Usenet news selection. Information filtering by ML techniques is widely discussed in the literature.

2.3.5. Hierarchical Categorization of Web Pages

Text Classification has recently aroused a lot of interest also for its possible application to automatically classifying Web pages, or sites, under the hierarchical catalogues hosted by popular Internet portals. When Web documents are catalogued in this way, rather than issuing a query to a general-purpose Web search engine a searcher may find it easier to first navigate in the hierarchy of categories and then restrict her search to a particular category of interest. With respect to previously discussed Text Classification applications, automatic Web page categorization has two essential peculiarities are: the hypertextual nature of the documents and the hierarchical structure of the category set.

2.4. Text Classification Problems

There are several difficulties that can restrain the process of categorizing texts, the main ones are the following [13] :

2.4.1. Redundancy

Redundancy and synonymy allow the expression of the same concept through different expressions, i.e., different ways of expressing the same thing. This difficulty is linked to the nature of the documents processed expressed in natural language as opposed to numerical data. "Lefèvre" illustrates this difficulty in the example of the cat and the bird: ”mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes”, which means my cat eats a bird, my big tomcat bites a sparrow and my favorite feline devours a little feathered animal. The same idea is represented in three different ways, different terms are used from one expression to another but ultimately it is the unfortunate bird that is being devoured by the cat.

2.4.2. Ambiguity

Unlike numerical data, the data textual texts are semantically rich because they are conceived and reasoned by human thought. Because of ambiguity, words are sometimes poor descriptors; for example, the word lawyer can refer to the fruit, the jurist, or even figuratively, the person who defends a cause.

2.4.5. Spelling

A term may contain spelling or typing errors as it can be written in several ways or written with a capital letter. This will affect the quality of the results. Because if a term is spelled in two ways in the same document (Tipaza/Tipasa, acknowlogemnt/acknowlogment, glamour/glamor grey/gray), simply searching for this term with a single graphical form omits the presence of the same term in other spelling.

2.4.4. Complexity of the learning algorithm

A text is usually represented as a vector containing the numbers of occurrences of terms in that text. However, the number of texts to be processed is very large. In addition, there is the number of terms in the same text. One can therefore get an idea of the size of the table (texts * terms) to be processed, which can only considerably complicate the task of classification by reducing the performance of the system.

2.4.5. Presence-Absence of a term

The presence of a word in the text indicates something the author intended to say. There is therefore a relationship involving the word and the associated concept, knowing full well that there are many ways of expressing the same thing. Therefore, the absence of a word does not necessarily imply that the concept associated with it is absent from the document. This sharp reflection makes it possible for us to leads us to be attentive to the use of self-learning techniques based on the exclusion of a particular word.

2.4.6. Compound words

The exclusion of compound words such as long-term, up-to-date, check-in, etc..., of which there is a very large number in all languages. Take the word up-to-date for example as 3 separate terms significantly reduces the performance of a system of classification nevertheless the use of the n-gram technique for the coding of texts considerably alleviates this problem of compound words.

3. Arabic Text Classification

The importance of Arabic Text Classification comes from the following main reasons; due to Historical, Geographical, Religious reason; Arabic language is a very rich with documents.

A study of the world market, commissioned by the Miniwatts Marketing Group. Shows that the number of Arab Internet users in the Middle East and Africa could jumped to 32 million in 2008 from 2.5 million in the year 2000, and in June 2012 this number jumped to more than 90 million users, the growth of Arab Internet users in the Middle East region (for the same period 2000-2012) is expected to reach about 2,640% compared to the growth of the world Internet users [19].

The big growth of the Arabic internet content in the last years has raised up the need for an Arabic language processing tools.

But on the other hand, there are many challenges facing the development of Arabic language processing tools including ATC tools. The first is that Arabic is a very rich language with complex morphology [19].

3.1. Arabic Text Classification Works

There are several research projects investigating and exploring the techniques in classifying English documents. In addition to English language there are many studies in European languages such as French, German, and Spanish and in Asian

languages such as Chinese and Japanese. However, in Arabic language there is little ongoing research in automatic Arabic document classification.

3.1.1. Al-Harbi et al.,

They attempts to attain a better understanding and elaboration of Arabic text classification techniques by using the aforementioned stages. The classification was performed on seven different datasets covering different genres and subject domains which are: Saudi Press Agency (SPA), Saudi Newspapers (SNP), Writers corpus, Islamic Topics and Arabic Poems. The performance of two well-known classification algorithms (SVM and C5.0) in classifying the seven Arabic corpora has been evaluated. The C5.0 algorithm surpass the SVM algorithm by about 10%; the SVM average accuracy is 68.65%, while the average accuracy for the C5.0 is 78.42% [20].

3.1.2. Meslah

He produced a text classification system for Arabic language documents. The achieved system uses 1) CHI statistics as a feature extraction method in the pre-processing step, and 2) Support Vector Machines (SVM) classification algorithm for text classification tasks. The corpus was gathered from online Arabic newspaper archives in addition to some other websites. This corpus contains 1445 documents classified into 9 categories. Experimental results indicate a high classification efficiency in term of F- measure compared to other classification algorithms which means that their X^2 feature extraction based on SVM classifier outperforms the Naïve Bayes and KNN classifiers; the SVM micro average F-score is 88.11%, while the F-score for the Naive Bayes is 84.54% and the F-score for the KNN is 72.72% [21].

3.1.3. Noaman et al.,

They presented the utilization of Naïve Bayes classifier with rooting algorithm to classify Arabic document. To approve the proposed algorithm, the authors created a corpus of 300 documents categorized into 10 classes. The corpus was collected from many newspaper articles gathered from various online newspaper websites. The experimental study demonstrates the achievement of the proposed classifier in terms

of error rate, recall measures, and accuracy, it accomplishes 62.23% of classification accuracy [22].

3.1.4. Goweder et al.,

The authors developed a Centroid-based technique for Arabic text classification. The proposed algorithm is evaluated using a corpus containing a set of 1400 Arabic text documents covering seven distinct categories. The experimental results show that the adapted Centroid-based algorithm is applicable to classify Arabic documents. The performance criteria of the implemented Arabic classifier achieved approximately figures of 90.7%, 87.1%, 88.9%, 94.8%, and 5.2% of Micro-averaging recall, precision, F-measure, accuracy, and error rates respectively [23].

Table 2.10 represent a small comparison on the previous works on Arabic Text Classification:

Table 2-10: Comparison between ATC works.

Research	Corpus	Accuracy
El-Harbi et al.	7 different datasets and each categorized into different classes.	SVM 68.65% C5.0 78.42%
Meslah	1445 documents categorized into 9 classes.	SVM 88.11% KNN 72.72% Naive Bayes 84.54%
Noaman et al.	300 documents categorized into 10 classes.	Naive Bayes 62.23%
Goweder et al.	1400 Arabic text documents categorized into 7 classes.	Centroid-Based Technique 94.8%

El-Harbi et al. used 7 different datasets with different classes using two models SVM and C5.0. The classifier C5.0 gave better accuracy than SVM.

Meslah used 1145 documents with 9 different classes using 3 different models: SVM, KNN and Naïve Bayes. The classifier SVM gave better accuracy than KNN and Naïve Bayes.

Noaman et al. worked on 300 documents with 10 different classes using only one model which is Naïve Bayes.

Goweder et al. works on 1400 Arabic text documents with 7 different classes using only Centroid-Based Technique.

3.2. Arabic Text Classification Problems

Developing text classification systems for Arabic documents is a challenging task due to the complex and rich nature of the Arabic language [20]:

- Particular characteristics of the Arabic language, e.g. ambiguity.
- The Arabic language has many grammatical forms, varieties of word synonyms.
- Different word meanings that vary depending on factors like word order and inclusion of diacritics.
- Different formulation and shapes for the same letter, based on the location of the letter in the word.

4. Conclusion

The growing use of the textual data which needs text classification to organize and extract pattern and knowledge from the documents specially in Arabic Language which is considered as a challenge in the automatic text classification because of greatness of the Arabic language and its ambiguities and difficulties.

In this chapter we studied the process of text classification which is divided into two different phases: training and classification. Then, we discussed two types of

text classification: supervised and unsupervised; along with the most important problems. Finally, we clarify the Arabic text classification; its works and difficulties.

Next chapter we will talk about poem classification, precisely the Arabic poems classification and almost everything relate to it: category, eras and topics of the poetry.

Chapter III:
Arabic Poems Classification

1. Introduction

Poetry has existed even before humans could read or write. It is said to be the freest literary genre. While all the other genres need characters, plot or narrative, poetry is free from all these restrictions. A poem can be as abstract or as specific as the poet needs it to be. The words in a poem may not be employed in their literal sense and may have a deeper contextual connotation.

Poetry, as a special form of literature, is crucial for computational linguistics. It has a high density of emotions, figures of speech, vividness, creativity, and ambiguity. Poetry poses a much greater challenge for the application of Natural Language Processing algorithms than any other literary genre [24].

In this chapter, we will talk about the importance of poetry and its types. Then, we move to the Arabic poetry eras and topics. Finally, we will talk about the Arabic poem classification works and difficulties.

2. The Poetry Importance

Through the years, poetry had gained a remarkable importance because [25]:

- It asks us to confront our humanity. It awakens our senses and develops our emotional awareness and intelligence. It allows us to discover truths we did not know we knew and the secrets of our own hearts. It allows us to see that poetry is the ordinary state of human thought and in it, we can confront representations of life and of ourselves. It allows us to connect to our interior spaces and come to find ourselves.
- Poetry matters because it develops an awareness and knowledge of language and its use through both reading and writing, gives us permission to work with language, develops skills of economy and precision that transfers to other writing and talking, develops oral and written skills and vocabulary, which is linked to higher achievement, and cultivates comprehension skills.

- Poetry matters because it demands and develops linguistic and metalinguistic awareness and knowledge, asking us to attend to the dialogue between and interrelationship among words and their features, content and rhetoric, the verbal and visual and auditory, and between content and form, pushing us toward Bereiter and Scardamalia's concept of composing as knowledge transformation.

3. Arabic Poetry

The Arabic poems are the earliest type of Arabic literature traditionally. It has always been at the heart of Arabic culture, not least as the oldest means for its earliest speakers to record their beliefs and wisdom, oral narratives and philosophy. It began in the Arabian Peninsula more than 1,500 years ago, predating Islam, and now has become global.

3.1. Arabic Poetry Types

Arabic poetry can be classified into two categories; classical and modern poetry. The second category shares more poetic features with modern poetry of other languages.

3.1.1. Classical Arabic Poetry

Classical Arabic poetry is considered as the origin of all Arabic poetry, from which other forms of Arabic poetry were derived. The poems in the classical Arabic poetry contain multiverses depending on the author and the purpose of the poem. Each verse consists of two parts of equal length; the first part is called "sadr" and the second part is called "ajuz". The term "buhur", which was invented by Al-Khalil bin Ahmed Al Farahidi, is used for metering the rhythmic system in a poem, and the measurement unit for "buhur/ بحور" is known as "Tafilah/ تفعيلة". In this type of poetry, the addition or removal of any letter from any verse will change the meter and the number of "Tafilah" in the verse. Therefore, this feature helps critics and scholars to identify the quality of an Arabic poem. Another important feature of the classical Arabic poetry is

the rhyme "qufiyah/ قافية". In general, all the verses in the poem have the same rhyme and these verses are described based on their number, as follows: one verse is called "Yetim/ يتيم", two and three verses are called "Natka/ نتفة", four, five, and six verses are called "Kuteah/ قطعة", while seven or more verses are called "Kassed/ قصيدة") [26].

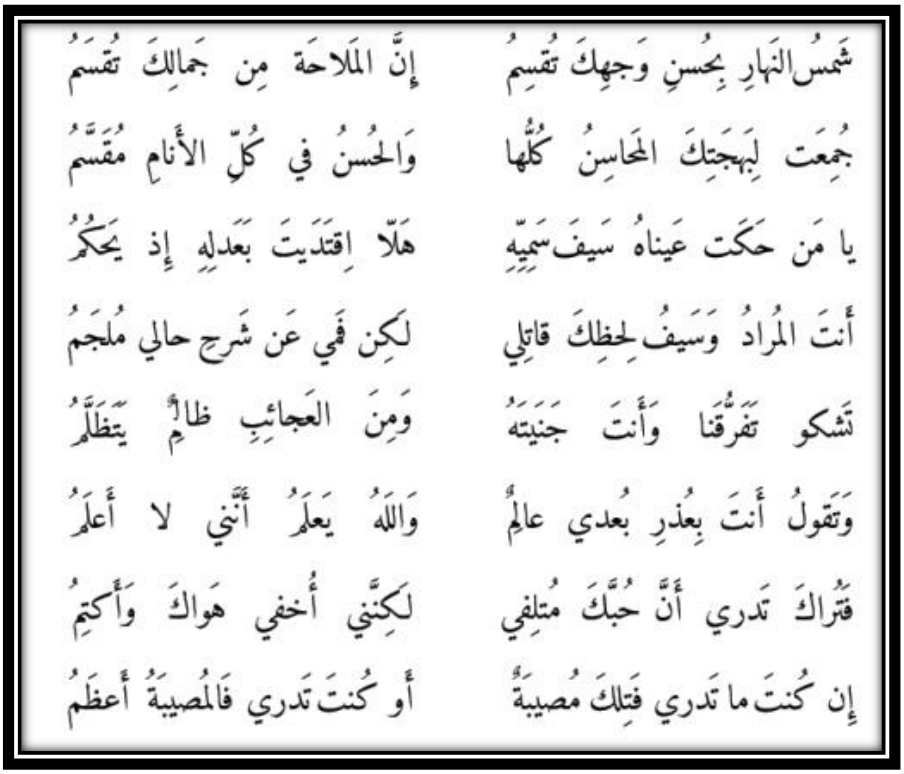

شَمسُ النَهارِ بِحُسنِ وَجهِكَ تُقسِمُ ... إِنَّ المَلاحَةَ مِن جَمالِكَ تُقسَمُ
جُمِعَت لِبَهجَتِكَ المَحاسِنُ كُلُّها ... وَالحُسنُ في كُلِّ الأَنامِ مُقَسَّمُ
يا مَن حَكَت عَيناهُ سَيفَ سَمَيِّهِ ... هَلّا اِقتَدَيتَ بِعَدلِهِ إِذ يَحكُمُ
أَنتَ المُرادُ وَسَيفُ لَحظِكَ قاتِلي ... لَكِن فَمي عَن شَرحِ حالي مُلجَمُ
تَشكو تَفَرُّقَنا وَأَنتَ جَنَيتَهُ ... وَمِنَ العَجائِبِ ظالِمٌ يَتَظَلَّمُ
وَتَقولُ أَنتَ بِعُذرِ بُعدي عالِمٌ ... وَاللَهُ يَعلَمُ أَنَّني لا أَعلَمُ
فَتُراكَ تَدري أَنَّ حُبَّكَ مُتلِفي ... لَكِنَّني أُخفي هَواكَ وَأَكتِمُ
إِن كُنتَ ما تَدري فَتِلكَ مُصيبَةٌ ... أَو كُنتَ تَدري فَالمُصيبَةُ أَعظَمُ

Figure 3-1: Classical Arabic Poetry example.

3.1.2. Modern Arabic Poetry

Typically, modern Arabic poems are written in consecutive short and uneven lines. Each line represents a verse. The lines can be arranged into sets of lines called stanzas/ مقطع. In some poems, all the stanzas contain the same number of lines. There is no limit on the number of lines in a poem. In many cases, the lines are arranged as separate paragraphs where a blank line is left between each two lines in the poem. It is also common that the entire poem is written in a single paragraph, or in several

paragraphs (stanzas) separated by numbers, blank lines, or punctuation marks. The structure and style in modern Arabic poetry is loose compared to classical poetry [27].

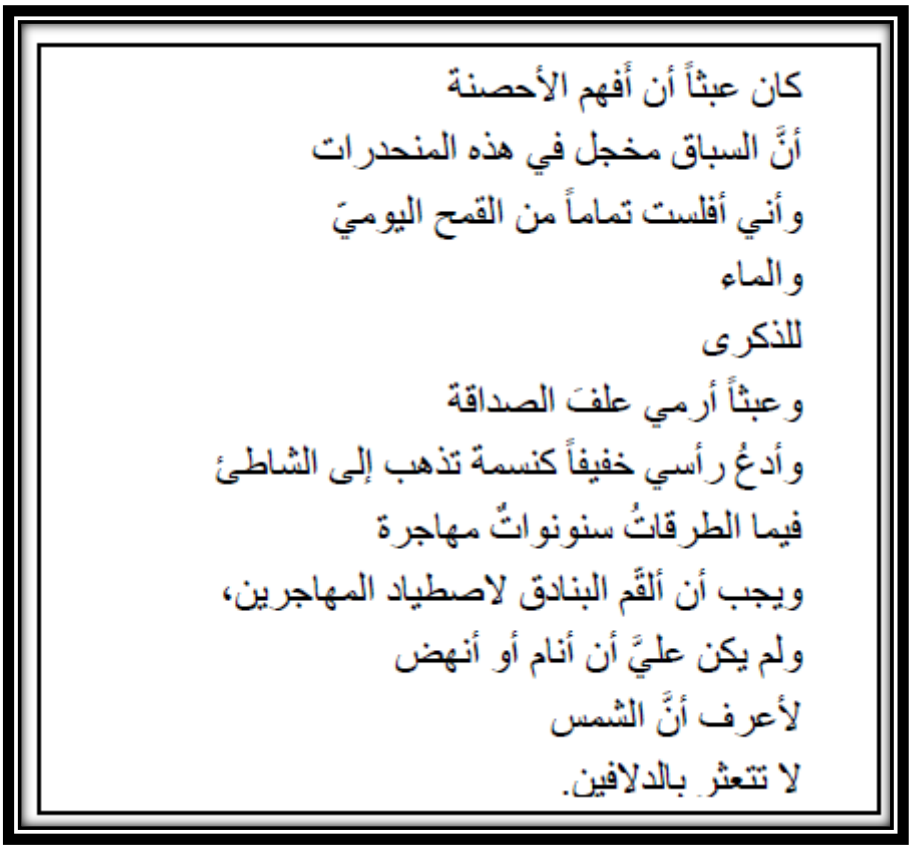

كان عبثاً أن أفهم الأحصنة
أنَّ السباق مخجل في هذه المنحدرات
وأني أفلست تماماً من القمح اليوميّ
والماء
للذكرى
وعبثاً أرمي علفَ الصداقة
وأدعُ رأسي خفيفاً كنسمة تذهب إلى الشاطئ
فيما الطرقاتُ سنونواتٌ مهاجرة
ويجب أن ألقّم البنادق لاصطياد المهاجرين،
ولم يكن عليَّ أن أنام أو أنهض
لأعرف أنَّ الشمس
لا تتعثر بالدلافين.

Figure 3-2: Modern Arabic Poetry example.

3.2. Arabic Poetry Eras

The poetry preserved the history of the Arabs. Poetry was their sole medium of expression, as they depicted a clear picture of human conditions, their experiences, battles, national achievements, glory of their rulers along with the special kind of wisdom, chivalry and valour etc. throughout the centuries. Generally the Arabic poetry is divided into the following eras:

3.2.1. Pre-Islamic Era العصر الجاهلي (500 to 622 A.D.)

The period before the writing of the Qur'an and the rise of Islam is known to Muslims as Jahiliyyah (الجاهلية) or period of ignorance. Here the word has been used as 'Jahiliyya' as the Arabs were unlettered and they did not have the power of reasoning and patience but they had the talent of exceptional caliber. They reached to perfection in poetry only through 'Oral' transmission and attained to its exactness during 'Jahiliyya' period preceding 7th century. That is to say before Islam and so it is called 'Pre-Islamic period'. There was no formal prose literature except folk tales, legends, proverbs, maxims etc. Poetry was the only medium to express their conditions and thoughts. In fact, their poetry represented all the aspects of their life and these are vividly reflected in the songs and odes of the poets which have come down to us. Their poetry carries the record of their life and culture, therefore, their poetry was regarded as 'Diwan al-Arab'الشعر ديوان العرب (Register of the Arabs) and a mirror of Arabian life. It was through poetry they expressed their sorrow and happiness of defeat and victory an expression of Arab people's cultural ideas and greatest aspiration [28]. Most famous Jahiliyyah poets:

- Abu Layla al-Muhalhel
- Antarah ibn Shaddad
- Imru' al-Qais
- Al-Khansa

3.2.2. Islamic Era العصر الاسلامي

It has begun with the migration of the prophet from Mecca to Medina in 622 A.D. Their Poetry literature consisted on Islamic feelings. Prophet of Islam and his four caliphs did not ignore poetry literature totally excepting that part which had inciting glorifying, satirical content. There is sufficient proof to show that he and his four caliphs liked poetry, as he asked 'Ata bin-al-Hadrami' and'Al-Khansa' to recite poetry. About a dozen of poets, who attached to the Prophet and sometimes praised him in simple terms. A tradition suggests that all the first caliphs were famous poets with Hazarat Ali at the top but they were not professional poets at all. They recited the

verses of other poets or their own only to lend vigor and poignancy to their words. From various sources it is seen that poetry literature was also in vogue along with the Quran and the Hadith literature in the early Islamic period. According to Ibn Khaldun, the most of the learned among the first Muslims who excelled in the religious or intellectual sciences were non-Arabs. At that time Arabs did not know the way by which learning is taught of the art of composing books and of the means where by knowledge is unregistered. Those who could repeat the Quran and relate the Hadith were called the readers. This oral transmission continued up to the days of Harun al-Rashid, who caused the Hadith to be set down in writing. A number of poetry during this period can be called Islamic because it contained ideas introduced by Islam such as the Unity of Allah, His power, majesty and glory or condemned pre-Islamic practices such as idolatry, superstitions and prejudices or advocated virtues recommended by Islam, such as fear of Allah, Unity, Restraint, Justice and Fair play [28]. Most famous Islamic poets:

- Hasan bin Thabit,
- Al-Hutai, Ka'ab bin Malik
- A'bdullah bin Rawa

3.2.3. Umayyad Era (661 to 750 A.D.)

The Umayyad period (661–750) is one of the most interesting and important for the critic of poetry. More than the verse of any other period prior to modern times, Umayyad poetry was in dynamic development and registered, obliquely and directly, the deeper changes in the spiritual condition of the times. This period of rapid development was flanked by more settled periods of poetic creativity: on the one side the pre-Islamic on the other the 'Abbasid poetry; and there can be no doubt that Umayyad poetry stems from a powerful poetic tradition of high achievement.

Umayyad poetry abounds with experiments. Many aspects of the poem were explored. New moods and themes were introduced, points of emphasis were shifted, and old motifs reappeared, intensified and sometimes exaggerated. This is a period in

which an unrivalled revolution took place spontaneously, unbound as yet by imposed traditionalism [29]. Most famous Umayyad poets:

- Al-Akhtal al-Taghlibi
- Al-Farazdaq
- Kuthayyir
- Bashar ibn Burd

3.2.4. Abbasid Era (750 to 1258 A.D.)

During the Abbasid period radical changes took place in poetry. There was a move towards shorter, less stylized poems with shorter meters. The poetry then reflected the civilized urban life with its opulent luxury and influence of foreign cultures mostly Persian. The Qasida was still an important form, but it was rarely of the traditional composite type. Rather it was a weighty mono-rhyme poem of reasonable length devoted to one of several recognized themes – eulogy, elegy and satire etc. The Abbasid rule proved itself the golden period of Muslim education. The first three centuries of the Abbasid period (750-1055AD) witnessed the great flowering of medieval Arabic literature and have been called its golden age, all this time Islamic culture assimilated major portions of the Greek and Roman intellectual traditions, adapted them and added its own distinctive contributions [28]. Most famous Abbasid poets:

- Abu Nuwas
- Abul Atahiya
- Al-Mutanabbi
- Abul-‘Ala al-Ma‘arri

3.2.5. Mamluks Era (1258 to 1516 A.D.)

Mamluk Arabic poetry presents often conflicting perspectives on religious life, exposing some of the complexity and centrality of competing religious views and their

underlying roots in Mamluk society. But despite their different emphases, many mystical and non-mystical religious poems from this period reveal a devotional quality, which is particularly pronounced in poems praising the Prophet Muhammad and his family. Though some such panegyrics were composed prior to the thirteenth century, it was under the Mamluks that a distinct poetic genre to praise the Prophet al-madih″ al-nabawi was extensively developed and codified by al-Busiri and his many imitators [28]. Most famous Mamluks poets:

- Al-Ashraf al-Ansari
- Al-Shabb al-Zarif
- Al- Busiri
- Safi al-din al-Hilli

3.2.6. Ottoman Era (1516 to 1798 A.D.)

The basic structural unit in Ottoman poetry is the couplet (Beyit). The basic characteristic of poetry is that poetical forms should be in meters called ‘aruz’, which is adopted from Persian literature. The vast majority of the diwan poetry was lyric in nature and the main genres in Ottoman poetry were ghazal (love), Qasida (panegyric) and mesnevi (romance). Here ghazal was the most common genre. Qasida was written for special occasions (like- birth, death, victory, enthronement, weddings and so forth). Ottoman diwan poetry is also characterized by the recurrence of three central figures: the lover, the beloved and the rival. Ottoman poetry is replete with symbolic relationships among these three. The lover may often be read as referring to the poet, whereas the beloved may be understood as referring to the sultan, a person in higher position or an actual beloved. The emotional situation of the lover was expressed with the use of metaphor and other literary devices such as simile. The classical ottoman poetry enriched by the widely respected work of oral poets [28]. Most famous Ottoman poets:

- Bâkî
- Fuzûlî
- Hayâlî

- Nedîm

3.2.7. Modern Era (1798 A.D. to present day)

The new trend of Romanticism did not come in a day, it had to undergo various changes and techniques by its composers. However, no one can deny the fact that in the development of Modern Arabic Poetry, the influence of West or rather to say European has been such that Modern Arabic Poetry deviated to some extend from its classical heritage. It is fact that the impact of West brought about a change in the Arab World not only in the technical military aspects, but also in the social, economic and literary fields. However, this impact of the West was felt much in literature and that was in poetry. From the literary point of view of Arabic poetry in Egypt in the early revival period gives the impression of adherence to traditional form with nostalgic feelings to past Arab greatness, which formed a natural preface to the Arab and owned them a respectable position in the modern world. The nostalgia is an inevitable element in revivalist poetry. The role of the poet, in illuminating poetry in modern literary renaissance is notable one. In the late 19^{th} and the earlier part of the 20th century much of the poetry was the product of particular events and situations. Basically, the poems were of the old forms and language with new themes [28]. Most famous Modern poets:

- Ahlam Mosteghanemi
- Mahmoud Darwish
- Nizar Qabbani
- Saadi Youssef

3.3. Arabic Poetry Topics

Topics may be termed as the overall unity of the Qasida. By this it meant the harmoniously integrated interaction between elements of symbolism, nostalgia and formal order, inherent in the standard pattern of Qasida with its varied modifications and other important poetic elements such as rhythm, diction and feeling. Therefore, it may say that, elements of the glorification of feat and arms, the praise of the virtues of

hospitality and the love, by far the richest element of the three, which has evoked some of the finest poems in the whole of Arabic literature.

3.3.1. Madih (المديح) (Eulogy / Panegyric)

Here the poet eulogies the bounty, the liberality and other pagan virtues of a chief who has helped him or his tribe in difficulty. It was not motivated, except in case of professional poets with the hope of getting a reward. The second category of eulogistic poetry has some resemblance to the type of love poetry inspired by a spirit of gallantry. It was the foremost significant form.

Example:

In the following verses, Qais Bel El-Malouh praised his Leila:

عَلَيكِ سَلامٌ لا سَلامَ مُوَدِّعٍ **** وَأَنتِ مِنى نَفسي وَأَنتِ سُرورُها

وَحُبُّكِ في الأَحشاءِ وَسطَ ضَميرُها **** فَحُبُّكِ في قَلبي مُقيمٌ مُصَوَّرٌ

فَأَنتُم مُنى قَلبي وَسُؤلي وَبُغيَتي **** وَأَنتُم ضِيا عَيني اليَمينِ وَنورُها

3.3.2. Hija (الهجاء) (Satire / Lampoon)

Here the poet satirizes his adversaries, as because insult was the main weapon and the target would be subject to withering ridicule. It was resorted to for defending one's honor or that of one's tribe and to expose the vices of the other party.

Example:

In the following verses, Iben Roumi insulted a humpback man which is kind of lampoon:

قصرت اخادعه وغاب قذاله **** فكأنه متربص لن يصفعا

وكأنما صفعت قفاه مره **** واحس ثانية لها فتجمعا

3.3.3. Fakhr (الفخر) (Self-glorification / Boasting)

Here the main task of the poet was to sing the qualities which he or his tribe possess, as like noble decent, bravery, revenge, chivalry, hops it ability, steadfastness to word and good neighborly relations in high esteem etc. for these he/she felt great pleasure and pride. However the Fakhr is gradually being transformed from private self-praise in matters of hasb/ nasb (Noble ancestry) to a medium whereby political struggle is glorified. The self-glorification occurs in the entire Muallaqat. It may be personal or tribal.

Example:

In the following verses, El-Mutanabi described his bravery and his eloquence:

أَلخَيْلُ وَاللَّيْلُ وَالبَيْداءُ تَعرِفُني **** وَالسّيفُ وَالرّمحُ والقرْطاسُ وَالقَلَمُ

صَحِبْتُ فِي الفَلَواتِ الوَحشَ منفَرِداً **** حتى تَعَجّبَ مني القُورُ وَالأكَمُ

يَا مَنْ يَعِزّ عَلَيْنَا أنْ نُفَارِقَهُمْ **** وَجدانُنا كُلَّ شيءٍ بَعدَكمْ عَدَمُ

3.3.4. Ritha (الرثاء) (Elegy)

Here the poet praises the qualities of a dead person. So, it was a combined sense of grief and consolation for loss with a rehearsal of the dead person's virtues that serve as an appropriate celebration of communal ideas. Among the elegy composers, Al-Khansa became celebrated as the greatest poetess of the Pre-Islamic period.

Example:

In the following verses, Al-Khansa praised her late brother:

أَبِنتُ صَخرٍ تِلكُما الباكِيَة **** لا باكِيَ اللَيلَةَ إلّا هِيَه

أَودى أَبو حَسّانَ واحَسرَتا **** وَكانَ صَخرٌ مَلِكُ العالِيَه

وَيلايَ ما أُرحَمُ وَيلاً لِيَه **** إذ رَفَعَ الصَوتَ النَدى الناعِيَه

3.3.5. Nasib (النسسسيب) (Verse on Beauty and Love of Women)

Here the poet portrays the passion of his heart. Described aptly as an elegiac reminiscence of love in which the poet expresses his gloomy and nostalgic meditations over the ruins of the desert encampment of the beloved. He also describes her charms and his own emotional reactions. In this kind of ode sometimes we find very interesting and vivid pictures of feminine beauty as well.

3.3.6. Wasf (الوصسسف) (Description/ descriptive poetry)

Here the poet gives the variety of description in a poem ranging from the simple enumeration of attributes. Sometimes poet also gives good descriptions of flowers and gardens. As in Muqallaqa of Imru al-Qais, the picture of his riding beast the scenery of the desert, its impressive solitudes, the exploits of romantic gallantry, the battle and the frolics method in the desert and so on. Later on, it was developed in the Abbasid period; thereafter in the Wasf pieces we find very little interaction between nature and the poet.

3.3.7. Ghazal (غزل) (Amatory verse)

The ghazal is a love lyric from five to 12 verses that probably originated as an elaboration of the qasida's opening section. The content was religious, secular, or a combination of both. It is a series of couplets, called shers, no more than a dozen or so. The Ghazal not only has a specific form but also traditionally deals with just one subject: love, specifically an unconditional and superior love. A traditional Ghazal consists of five to fifteen couplets, typically seven. Essentially it was a new development in the Arabic poetry, emerged at the early days of the community of Muslim.

Example:

In the following verse, Imru Al-Qais described his amatory towards a woman:

أَمِن ذِكرِ سَلمى إِذ نَأَتكَ تَنوصُ **** فَتَقصِرُ عَنها خُطوَةً وَتَبوصُ

وَكَم دونَها مِن مَهمَهٍ وَمَفازَةٍ **** وَكَم أَرضُ جَدبٍ دونَها وَلُصوصُ

3.3.8. Hamasa (حماسسسة) (Bravery and fortitude/War poetry)

One of the great anthologies of Arabic literature. It was gathered together in the 9th century. It did not only mean bravery in war but a lot besides this, it meant a resolute and unyielding attitude towards the forces of nature whatever their form.

3.3.9. Al-Hikma (الحكمة) (Wise sayings)

The verses which contains some moral truth or precepts of practical wisdom. Elegiac verses often replete with reflections on life and death which lead on to the utterance of the words of wisdom as for instance Labid bin Rabia, the Muallaqa poet says in a line of his elegies.

4. Arabic Poetry Classification

Arabic Poetry Classification is a technique subset from Text Classification. We can classify the Arabic Poetry in too many way such as: according to eras, to topics and to rhythms. It processing is similar to Text Classification ones. It still causes a major problem to researches because of the challenging Arabic language.

4.1. Arabic Poetry Works

All the works of Arabic poetry are recent and because of the lack of the publicly Arabic corpora is still remain a difficult task to the researches. Here it is some of those works.

4.1.1. Al-Falahi et al (2017)

Before this research, no published works and researches about authorship attribution in Arabic poems, in this work they used both of NB and SVM algorithms in the old Arabic poetry context. The poetry corpus includes seventy-three poets with

18646 Qasidah, the full words are equal 1235402. Those words were divided into 1856436 words for training dataset and 106546 words for the testing dataset. The maximum accuracy value is 98.63% of true attribution with apply SVM, and a maximum accuracy equal to 97.26 % by NB [30].

4.1.2. Almuhareb et al (2013)

On this work, they presented a novel method for recognizing and extracting classical Arabic poems found in textual sources. The method used was the common features of classical Arabic poems such as structure, writing style, and rhyme; and applied them in the recognition process [27].

4.1.3. Alsharif & Ghneim (2013)

The corpus contains 1231 Arabic poems that vary in length in four main affect categories (Retha, Ghazal, Fakhr, Heja). Alkhalil was used for stemming and rooting, Khoja for rooting, Stanford POS tagger, and Alkhalil to extract words by their tags and WEKA to apply the machine learning algorithms; four main machine learning algorithms are compared: Naïve Bayes, SVMs, VFI and Hyperpipes. The best precision achieved was 79% using Hyperpipes with non-stemmed, non-rooted, mutually deducted feature vectors containing 2000 features [31].

4.1.4. Alnagdawi and Rashideh

They proposed a context-free grammar-based tool for finding the poem meter name. The proposed tool was worked only with trimmed Arabic poems (words with diacritics “Tashkeel”) [30].

4.1.5. Al Hichri and Al Doori

Their work was an expert system for classifying Arabic poems depending on rhythmic structure of short and long syllables. They used a rule-based algorithm and it is applied in several passes. Then they converted the resulting strings into binary sequence. Finally, they calculated a distance measure between the binary pattern of a

verse in the unknown poem and the binary patterns of all poetry seas. They considered their experimental results successful [32].

4.1.6. Iqbal AbdulBaki Mohammad (2009)

They used Naïve Bayes for the classification and Al-Shalabi, Kanaan, and AlSerhan algorithm for extracting roots of the words in any poem. The corpus contains twenty poems for the training set with six verses and twenty poems for the testing set. The best accuracy was 25% [32].

4.1.7. Ahmed ZEGGADA Rabah MOULAI (2019)

They developed a system of classifying Arabic poems according to the periods in which they were written. The best results they achieved by using the Multinomial Naive Bayes (MNB) algorithm, with an accuracy equal to 70.21%, and F1-Score of 68.8% and a Kappa equal to 0.398, without filtering stop words with corpus contain more than 30k poems [13].

Table 3-1: Comparison between APC Works.

Work	Author/Year	Description
Machine Learning for Authorship Attribution in Arabic Poetry.	Al-Falahi Ahmed, Ramdani Mohamed, and Bellafkih Mostafa (2013)	A corpus with seventy-three poets with 18646 Qasidah, the full words are equal 1235402 these words are divided into 1856436 words for training dataset and 106546 words for the testing dataset, using NB and SVM for the classification. The maximum accuracy was 98.63% (SVM) and 97.26 % (NB).

Recognition of Classical Arabic Poems.	Abdulrahman Almuhareb Ibrahim Alkharashi Lama AL Saud Haya Altuwaijri (2013)	A novel method for recognizing and extracting classical Arabic poems, utilizes the common features of classical Arabic poems such as structure, writing style, and rhyme.
Emotion Classification in Arabic Poetry using Machine Learning.	Ouais Alsharif, DeemaAlshamaa, Nada Ghneim (2013)	The corpus contains 1231 Arabic poems in four main affect categories (Retha, Ghazal, Fakhr, Heja), and four machine learning algorithms was used (NB, SVM, VFI and Hyperpipes). The best precision achieved was 79% using Hyperpipes.
Classifying Arabic poems depending on rhythmic structure.	Al Hichri and Al Doori	An expert system for classifying Arabic poems depending on rhythmic structure of short and long syllables, they used a rule-based algorithm, and considered their experimental results successful.
NAIVE BAYES FOR CLASSICAL ARABIC POETRY CLASSIFICATION.	Iqbal AbdulBaki Mohammad (2009)	The corpus contains twenty poems for the training set with six verses and twenty poems for the testing set, the best accuracy was 25%.
CATEGORISATION AUTOMATIQUE DES TEXTES ARABES.	Ahmed ZEGGADA and Rabah MOULAI (2019)	The best results they achieved by using the Multinomial Naive Bayes (MNB)

		algorithm is accuracy equal to 70.21%, F1-Score of 68.8% and a Kappa equal to 0.398, without filtering stop words with corpus contain more than 30k poems.
Finding Arabic poem meter using context free grammar.	Alnagdawi and Rashideh (2013)	A context-free grammar-based tool for finding the poem meter name. The proposed tool worked only with trimmed Arabic poems.

To identify authorship attribution, Al-Falahi et al, used a corpus with 73 poets and 18646 poems by using NB and SVM classifiers. Their results is that SVM gave better accuracy than NB.

Abdulrahman et al, used a novel method for recognizing and extracting classical poems by using common feature of it such as structure, writing style and rhythm.

Ouais Alsharif et al, used a corpus with 1231 Arabic poems in four main categories Retha, Ghazal, Fakhr and Heja by using four algorithms NB, SVM, VFI and Hyperpipes. The last one gave the best precision.

Al-Hichri and Al-Doori, used an expert system for classifying Arabic poems depending on rhythmic structure of short and long syllables by using a rule-based algorithm. Their experimental results successful.

Iqbal AbdulBaki, used a corpus with 20 poems for training set and 20 poems for testing set. His best accuracy was 25%.

Alnagdawi and Rashideh, used a context-free grammar based tool for finding the poem meter name and their tool worked only with trimmed Arabic poems.

Ahmed ZEGGADA and Rabah MOULAI, used a corpus with more than 30k poems. Their best results they achieved by using the MNB algorithm: accuracy (70%), F1-Score (68%) and Kappa (0.398) without filtering stop words.

4.2. Arabic Poetry Difficulties

Despite the rapid progress in this area in some international languages, the analysis of classical Arabic poetry has not received a sufficient attention due to the difficulty of the Arabic language, and the difficulties of analyzing its poetic theories. There are the most common problems:

- The convoluted presence of emotions throughout the body of a poem. Unlike poets of the modern days old poets were inclined not to display emotion explicitly, but rather, through complex, convolved structures. These complex structures pose a hurdle to classification methods trained on language tokens since they do not take into account higher level language structures such as semantics and pragmatics where most of the emotional information lies.
- Cannot really define the topic of the poems unless you fully read and understand the poem completely.
- Poets use too many metaphor to really define to real meaning especially in the classical poems.
- Automatic Rhythm Analysis detection and finding the number of rhythm of the verse in poems.
- Variety of the Arabic words.
- Scansion which marks the metrical pattern of a poem by breaking each line of verse up into feet[1] and highlighting the accented and unaccented syllables.

[1] Foot is the basic unit of measurement in poetry.

5. Conclusion

There is no doubt that Arabic poetry is now passing through an acute phase of experimentation. Even more fundamental experiments in this poetry are yet to be made. Everything in current Arab life is dynamic, and despite the fact that modern Arabs are now decisively oriented towards technology, poetry still plays an important role in their culture and one feels, will once again prove to be the first medium of expression of a quick changing sensibility. These changes will be concentrated at the beginning on the intricate and highly varied metric forms of the Arabs where the real adventure lies.

In the further chapter, we will see in details the global architecture of our proposed solution.

Chapter IV: Conception & Modelization of the Proposed Solution

1. Introduction

Like we have seen in the previous chapters, Arabic poetry caused a major challenge for the researchers and to solve this problem we proposed a comparative classifiers to train, test and evaluate.

In this Chapter, we will present the modelization of our proposed solution. We will start by a remainder of the problematic of our project. Then we are going to present the global architecture of our system. After that, we will define our approach for text classification which consist two stages: pre-processing and choosing the best model between eight to apply it in Arabic Poem Classification.

2. Proposed Solution

Arabic Text classification has different problems because of the nature of the Arabic language and Poetry is very challenging form of the Arabic text and especially in the meaning; the poet not only express his feeling and his emotions but also his environment in very different ways according to their eras and poem topics.

The researchers found poetry very hard task to deal with especially the foreign ones. In our case we are interested practically in the Arabic poem classification according to their Eras and to their topics.

The problem of automatic text categorization is mainly solved by automatic learning. For this purpose, several algorithms developed for any problem in machine learning have been adapted and applied in our field of research. The aim is to find a functional link, also called a prediction model, between the texts to be classified and the set of categories.

Saline Classification is our proposed solution to the Arabic poems classification problem. We trained, tested and evaluated a several algorithms and then we choose the best one to predict poems' category, era and topic.

3. Global Architecture

We present here, the functional architecture of our categorization system.

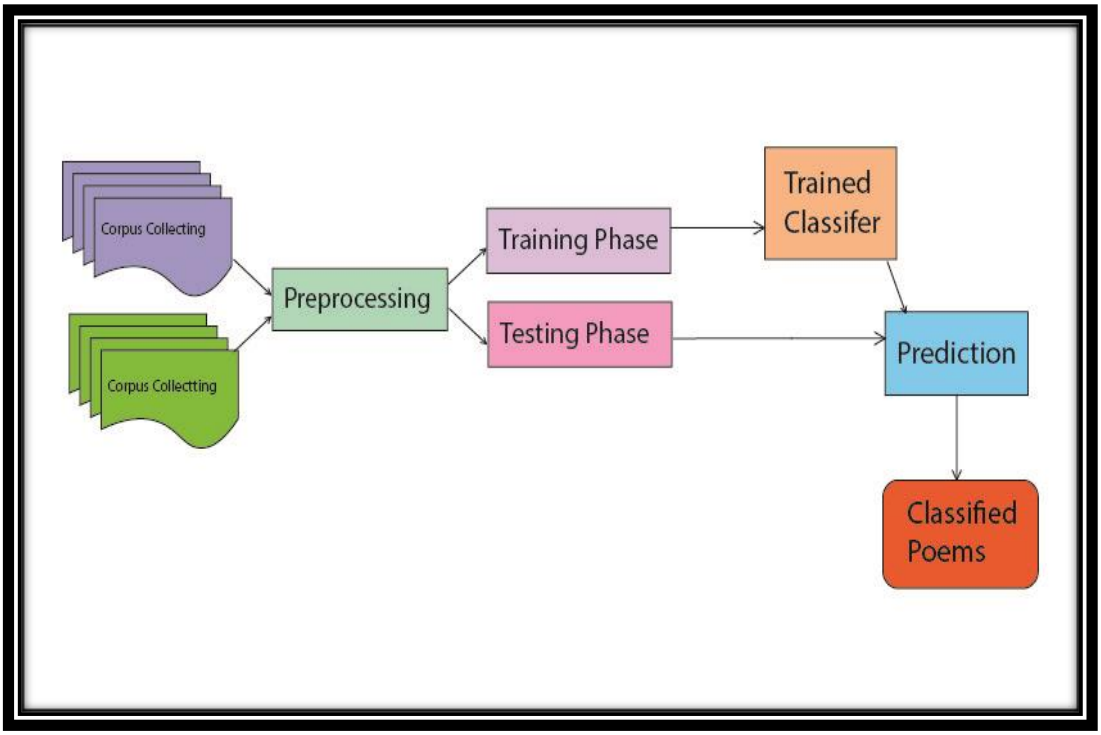


Figure 4-1: Global Architecture.

Our global architecture contains five stages: collecting corpora phase, preprocessing phase, training phase that give us a trained classifier as a result. Then, we test and evaluate it in the testing phase. Finally, in the prediction phase we get a classified poems as a final result.

3.1. Corpora

As mention in chapter two, one of the difficulties that encounter this work and other researches in the field of Arabic linguistics was the lack of publicly available Arabic corpus for evaluating text categorization algorithms.

There is a need for a freely-accessible corpus of Arabic. There are no standard or benchmark corpora. All researchers conduct their researches on

their own compiled corpus. Arabic language is highly inflectional and derivational language which makes text mining a complex task. In Arabic TC research field, there are some published experimental results, but these results came from different datasets, it is hard to compare classifiers because each research used different datasets for training and testing.

3.1.1. Corpora Building Steps

The main consecutive phases of building a text classification system (presented in figure 4-2) has been described in chapter two. The first phase in construction process is to build a text dataset which involves compiling and labeling text documents into corpus. We collect poems from two websites adab.com and aldiwan.net using the open source offline explorer, HTTrack[1] . The process also includes extract the information needed to build the corpus using html files and web scraping. The final step is to save the structured data into excel file. The following figure sums up these steps.

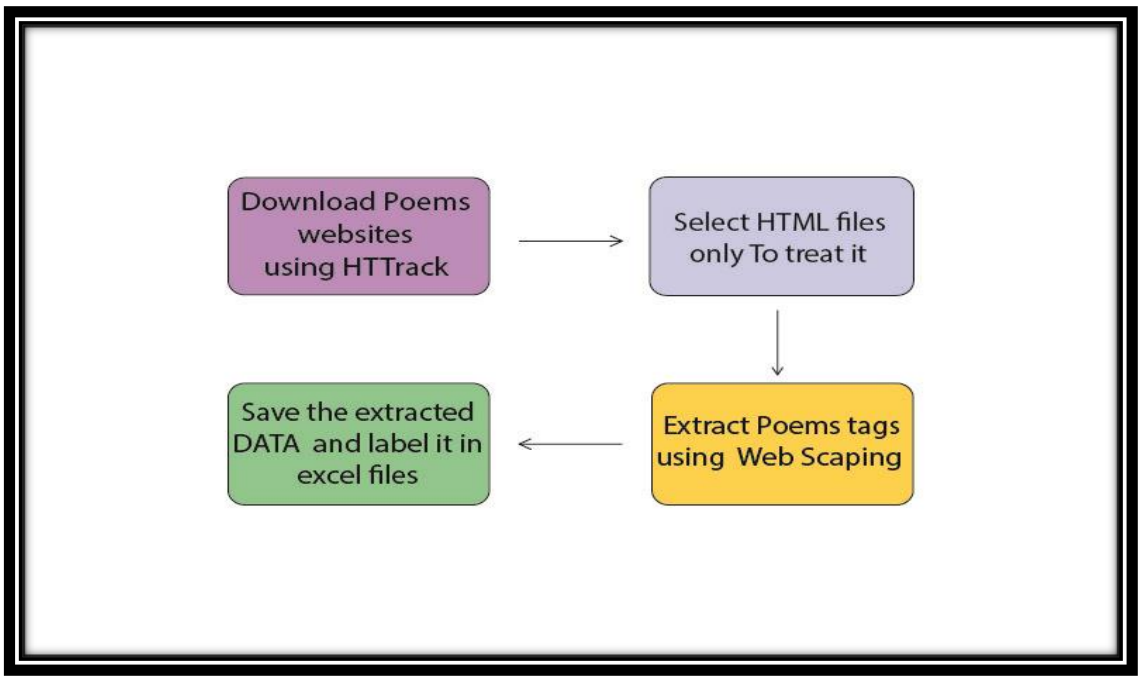


Figure 4-2: Corpus Building Step.

[1] www.httrack.com

As we see in *Figure 4-2*, the collecting of Corpora takes place in four stages staring with extracting the poems from the websites and finishing with a structured Excel Data.

3.1.1.1. Poems Websites

In this step, we selected the websites that we had downloaded it to extract poems from it later on using HTTrack like it show in the figure 4-3.

HTTrack is a free software and easy-to-use offline browser utility. It allows you to download a World Wide Web site from the Internet to a local directory, building recursively all directories, getting HTML, images, and other files from the server to your computer. HTTrack arranges the original site's relative link-structure. Simply open a page of the "mirrored" website in your browser, and you can browse the site from link to link, as if you were viewing it online.

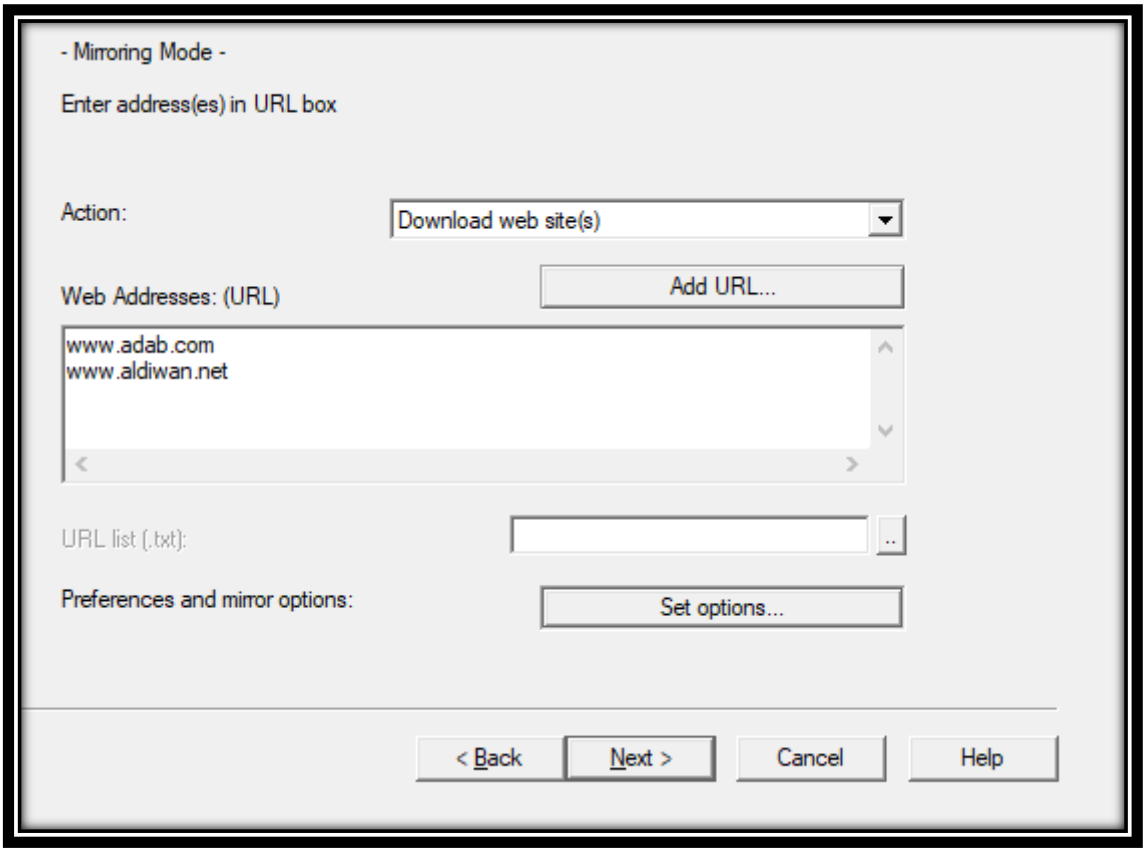


Figure 4-3: HTTrack website copier.

3.1.1.2. HTML Files Selection

As we said in the first step, HTTrack not only downloaded the html files but also images, xml files and other files from the server. That is why in this step, we needed to select only the html files and save into a folder to use later it to extract the poem tags.

3.1.1.3. Web Scraping

Web Scraping is a technique employed to extract a large amount of data from websites using "Beautiful Soup" by the HTML Tags and also using UTF-8.

Beautiful Soup is a Python library for parsing structured data**.** It allows you to interact with HTML in a similar way to how you would interact with a web page using developer tools.

- **Poet Extraction**

Here we extract the name of the poet for both corpus

For Adab corpus:

```
poetTitle = all_content[1].find("title").text
#poet extraction
poet=poetTitle.split('..')[1].split(':')[0]
```

Figure 4-4: Poet Extraction for Adab corpus.

For Aldiwan corpus:

```
 poet extraction
poet= all_content[1].find("meta",attrs={"name": "author"}).get('content')
```

Figure 4-5: Poet Extraction for Aldiwan corpus.

- **Title Extraction**

Here we extract the title of the poem for both corpus.

For Adab corpus:

```
poetTitle = all_content[1].find("title").text
#title extraction
title=poetTitle.split(':')[1]
```

Figure 4-6: Title Extraction for Adab corpus.

For Aldiwan corpus:

```
title extraction
title= all_content[1].find("meta",attrs={"name": "twitter:title"}).get('content').split('-')[0]
```

Figure 4-7: Title Extraction for Aldiwan corpus.

- **Poem Extraction**

Here we extract poem for both corpus.

For Adab corpus:

```
#poem extraction
poem = [item.get_text(strip='True')
   for item in all_content[70]('p', attrs={'class': 'poem'})][:-2]

quasida =[]
#i howa indice w b howa text (elment)
for i,b in enumerate(poem):
        quasida.append(poem[i])
```

Figure 4-8: Poem Extraction for Adab corpus (single versed poems).

```
#poem extraction
poem1 = [item.get_text(strip='True')
   for item in all_content[70]('p', attrs={'class': 'poem'})][:-2]
poem2 = [item.get_text(strip='True')
   for item in all_content[70]('font', attrs={'class': 'poem'})][:-2]

sader =[]
ajez =[]
#i howa indice w b howa text (elment)
for i,b in enumerate(poem2):
    if i%2==0:
        ajez.append(poem2[i])
    else:
        sader.append(poem2[i])
```

Figure 4-9: Poem Extraction for Adab corpus (two versed poems).

For Aldiwan corpus:

```
#poem extraction for one column
        poem = soup.find('h4').text
        quasida=[]
           #i howa indice w b howa text (elment)
        for i,b in enumerate(poem):
               quasida.append(poem)
```

Figure 4-10: Poem Extraction for Aldiwan corpus (single versed poems).

```
poem extraction for two columns
selector2='div.row h3'
found=soup.select(selector2)
poem = [item.text.strip()  for item in found][:-1]
sader =[]
ajez =[]
#i howa indice w b howa text (elment)
for i,b in enumerate(poem):

    if i%2==0:
        sader.append(poem[i])
    else:
        ajez.append(poem[i])
```

Figure 4-11: Poem Extraction for Aldiwan corpus (two versed poems).

- **Category Extraction**

  Here we extract category for Adab corpus by using the following selector 'td'.

- **Era Extraction**

  Here we extract eras for both corpus.

  For Adab corpus:

```
#Era extraction
selector=' td a '
found=soup.select(selector)
Era = [item.text  for item in found][8]
```

Figure 4-12: Poet Extraction for Adab corpus.

For Aldiwan corpus:

```
era extraction
Era = soup.find('h2')
Era = Era.text.split('»')[-3]
```

Figure 4-13: Poet Extraction for Aldiwan corpus.

- **Topic Extraction**

```
topic exctraction
selector='div.col a '
found=soup.select(selector)
topic = [item.text  for item in found][0]
```

Figure 4-14: Topic Extraction for Aldiwan corpus.

After the extraction of those information, we put them into a data frame and transformed it to an excel file.

3.1.1.4. Structured dataset

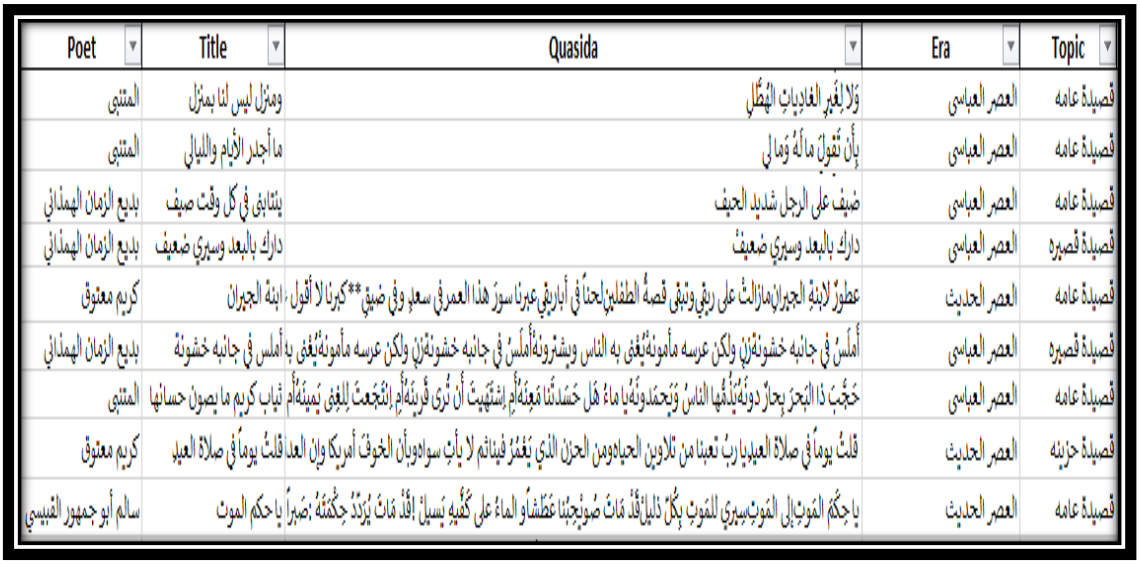
The data obtained from the three steps above we used it to build our structured dataset and saved it into an excel file. The poem information that we extracted and used to build this structured dataset are: “Poet”, ”Title”, ”Quasida” for single versed pomes, “Sader” and “Ajez” for two versed poems, ”Category” for Adab corpus, ”Era” and ”Topic” for Aldiwan corpus. We have 4 excel files which means 4 structured dataset as shown in the following figures.

Poet	Title	Quasida	Era	Category
صلاح جاهين	على رجلى دم .. نظرت له ما احتملت	على رجلى دم .. نظرت له ما احتملت‌على	العصر الحديث	Folk
عباس جيجان	سياستنا	بس احنا سياستنا ذبح وارهاب من الباب	العصر الحديث	Folk
جوزيف حرب	ورقو الأصفر	ورقو الأصفر شهر أيلول تحت الشبابيك	العصر الحديث	Folk
فائق عبدالجليل	آخر زيارة	آخر زيارة .. حسيت في آخر زيارةشي بع	العصر الحديث	Folk
جوزيف حرب	ظلّك كذاب !	ظلك، كذاب.. ما بيحسّ.. مَ بتفرق معو	العصر الحديث	Folk
عباس جيجان	العراق	ما أنت بعيد أنت العين وأقرب من الجفـ	العصر الحديث	Folk
بدر الصفوق	مذرحه	مذرحه 1شف إذا ضاق المكان وصرت م	العصر الحديث	Folk
عبدالرحمن الأبنودي	الإمام	إلى فؤاد حدادإنت الإمام الكبير .. وأصا	العصر الحديث	Folk
جوزيف حرب	ما قدرت نسيت	زعلى طول أنا وياك و سنين بقيتجرب	العصر الحديث	Folk

Figure 4-15: Adab dataset single versed poems.

Poet	Title	Sader	Ajez	Era	Category
راشد الخضر	ركن الشعر	طاح ركن الشعر والفناعنا ش	ركن الشعرحزن عم الكون	العصر الحديث	Folk
متعب التركي	قالوا حبيبك	قالوا حبيبك يسمع الشعر	قالوا حبيبكيقراه بعيوني قم	العصر الحديث	Folk
عمر الفرّا	عودي	عودي إلي . .أجن في ملقاك	عودي‌وأعيش فيك عبادة	العصر الحديث	Folk
خالد الردادي	نص دّ/ شعبى عتيق	طال السكون وعسعس اللي	نص دّ/ شعبى عتيق‌تاه النج	العصر الحديث	Folk
راشد الخضر	من قلبى	دشيت باب الحب بس مار	من قلبى‌بيت بخرج ولا ربت	العصر الحديث	Folk
محمد الأحمد السديري	يا فزعة المظيوم	يا مرحبا حييت يا عز الاقرا	يا فزعة المظيوم‌واهلا بشو	العصر الحديث	Folk
حمد بن على الكعبى	راية العز	رفرفي يا راية العز من فوقا	راية العز‌وأشعلى حب الوط	العصر الحديث	Folk
سعد بن علّوش	صبح الرشاش	بغيت أأضمّك بصدري على	صبح الرشاش‌عف الله عن	العصر الحديث	Folk
خالد الردادي	الهندول	يا ياسمين من اليمين ومن شـ	الهندول‌و(اربعينية) على يا	العصر الحديث	Folk

Figure 4-16: Adab dataset two versed poems.



Poet	Title	Quasida	Era	Topic
المتنبي	ومنزل ليس لنا بمنزل	وَلا لِغَيرِ الغَادِياتِ الهُطَّلِ	العصر العباسي	قصيدة عامه
المتنبي	ما أجدر الأيام والليالي	بِأَن تَقولَ مالَهُ وَما لي	العصر العباسي	قصيدة عامه
بديع الزمان الهمذاني	يتنابى في كل وقت صيف	ضيف على الرجل شديد الحيف	العصر العباسي	قصيدة عامه
بديع الزمان الهمذاني	دارك بالبعد وسيري ضعيف	دارك بالبعد وسيري ضعيف	العصر العباسي	قصيدة قصيره
كريم معتوق	ابنة الجيران	عطورٌ لابنةِ الجيرانِ مازالتْ على ريقي وتبقى قصةُ الطفلينِ لحناً في أباريقي عبرنا سورَ هذا العمرِ في سعدٍ وفي ضيقِ **كبرنا لا أقول :	العصر الحديث	قصيدة عامه
بديع الزمان الهمذاني	أنا أمس في جانبه خشونة	أَمْلَسُ في جانبِهِ خشونةٌ لِي ولكن عرسه مأمونةٌ يُغْنِي بِهِ الناس ويشترونهُ أَمْلَسُ في جانبه خشونةٌ لِي ولكن عرسه مأمونةٌ يُغْنِي	العصر العباسي	قصيدة قصيره
المتنبي	ثياب كريم ما يصون حسانها	خَجِّبْ ذا البَحرَ بِحارٌ دونَهُ لَيْلُهَا الناسُ وَيَحمَدونَهُ يا ماءُ هَل حَسَدتَنا مَعونَهُ أَم إِشتَهَيتَ أَن تُرى فَرِيئَةً أَم إِنتَجَعتَ لِلغِنى يَمينَهُ أَم	العصر العباسي	قصيدة عامه
كريم معتوق	قلت يوماً في صلاة العيد	قلتُ يوماً في صلاةِ العيدِ يارب تعبنا من تلاوين الحياةومن الحزن الذي يَغمُرُ فينا نم لا يأتِ سواهوبأن الخوفَ أمريكا وإن العد	العصر الحديث	قصيدة حزينه
سالم أبو جمهور القيسي	صبراً يا حكم الموت	يا حِكَمَ المَوتِ إِلى المَوتِ سِيري لِلمَوتِ بِكُلِّ ذَليلٍ قَد مَاتَ صُوتُجِبنا عَطشاً و الماءُ على كُفَّيهِ يَسيلُ قَد مَاتَ يُرَدِّدُ حِكْمَتَهُ :	العصر الحديث	قصيدة عامه

Figure 4-17: Aldiwan dataset single versed poems.

Poet	Title	Sader	Ajez	Era	Topic
المتنبي	صلة الهجر لي وهجر الوصال	صِلَةُ الهَجرِ لي وَهَجرُ الوِصالِ فَقَدا الجِ	نَكَساني في السُقمِ نُكسَ الهِلالِ يُقصُ مِنهُ يَز	العصر العباسي	قصيدة عامه
المتنبي	أبعد نأي المليحة البخل	أَبعَدُ نَأيِ المَليحَةِ البُخلُ مَلولٌ ما يَدومُ	في البُعدِ ما لا تُكَلِّفُ الإِبلُ مِن مَلَلٍ دائِمٍ بِها	العصر العباسي	قصيدة عامه
المتنبي	ليس هم ارتحالا	وَحُسنَ الصَبرِ زَمّوا لا الجِمالاتَهَيَّبَني فَفاجَأَ	بَقائي شاءَ لَيسَ هُمُ إِرتِحالا تَوَلَّوا بَغتَةً	العصر العباسي	قصيدة عامه
المتنبي	في الخد أن عزم الخليط رحيلا	مَطَرٌ تُزيدُ بِهِ الخُدودُ مُحولا في الخَدِّ أَن عَزَمَ الخَليطُ رَحيلا	يا نُظرَةً دَ	العصر العباسي	قصيدة عامه
المتنبي	أرى حللا مطواة حسانا	عَدا بي أَن أَراكَ بِها اِعتِلالي أَتُطوي ما عَلَيكَ هِ	أَرى خُلَلاً مَطَوّاةً حِساناً وَهَبكَ طَوَيتَها وَ	العصر العباسي	قصيدة عامه
المتنبي	عذلت منادمة الأمير عواذلي	في شُربِها وَكَفَت جَوابَ السائِلِ وَحَمَلتُ شُرَّ	عَذَلَت مُنادَمَةُ الأَميرِ عَواذِلي مَطَرَت سَ	العصر العباسي	قصيدة عامه
المتنبي	بدر فتى لو كان من سؤاله	يَوماً تَوَفَّرَ حَظُّهُ مِن مالِهِ وَيَقِلُّ ما يَأتيهِ في إِقـ	بَدرٌ فَتىً لَو كانَ مِن سُؤّالِهِ تَتَخَيَّرُ الأَفعالُ	العصر العباسي	قصيدة عامه
المتنبي	قد أبتُ بالحاجة مقضية	وَعِفتُ في الجَلسَةِ تَطويلَها خَيرٌ لِنَفسي مِن	قَد أَبتُ بِالحاجَةِ مَقضِيَّةً أَنتَ الَّذي طو	العصر العباسي	قصيدة قصيره
المتنبي	لك يا منازل في القلوب منازل	أَقفَرتِ أَنتِ وَهُنَّ مِنكِ أَواهِلٌ أَولاكُما بِبُكى	لَكِ يا مَنازِلُ في القُلوبِ مَنازِلٌ يَعلَمنَ ذا	العصر العباسي	قصيدة عامه
المتنبي	أمانكم من قبل موتكم الجهل	وَجَزَّكُمُ مِن جِفَّةٍ بِكُمُ النَملُ فَطِشتُم إِلى الدَعو	أَمانُكُمُ مِن قَبلِ مَوتِكُمُ الجَهلُ وَلَيدَ أَيِّ	العصر العباسي	قصيدة عامه
المتنبي	يا أكرم الناس في الفعال	وَأَفصَحَ الناسِ في المَقالِ فَهَكَذا قُلتَ في النَوا	يا أَكرَمَ الناسِ في الفَعالِ إِن قُلتَ في ذا المَ	العصر العباسي	قصيدة قصيره
المتنبي	أتاني كلام الجاهل ابن كيغلغ	يَجوبُ حُزوناً بَينَنا وَسُهولاوَيَبغي سِوى رُمحٍ	أَتاني كَلامُ الجاهِلِ اِبنِ كَيَغلَغ وَلَو لَم يَكُ	العصر العباسي	قصيدة هجاء
المتنبي	لا تحسبوا ربعكم ولا طلله	أَوَلَ حَيٍّ فِراقُكُم قَتَلَه وَأَكثَرَت في هَواكُمُ العَذَ	لا تَحسَبوا رَبعَكُم وَلا طَلَلَه قَد تَلِفَت قَبلَ	العصر العباسي	قصيدة غزل

Figure 4-18: Aldiwan dataset two versed poems.

3.1.2. Corpora Summary

We used various corpora to perform our experimentations, the corpora variations include small/large size corpus, with few and more categories. We collected two corpora, we collect them from: Adab and Aldiwan website and each corpora have two other corpora.

3.1.2.1. Adab Corpus

Adab corpus contains 2486 single versed poems and 2163 two versed poems. Each poem belongs to 1 of the 5 existent eras and to 1 of the 3 existent categories.

Table 4-1: Adab Corpus Statistic according to eras.

Eras	Single-versed Poems	Two-versed Poems	Total
العصر الجاهلي	0	109	109
العصر الإسلامي	0	135	135
العصر العباسي	0	143	143
العصر الأندلسي	0	70	70
العصر الحديث	2486	1706	4192

Table 4-2: Adab Corpus Statistic according to categories.

Category	Single-versed Poems	Two-versed Poems	Total
Fassih	1735	1383	3118
Folk	212	759	971
World	539	0	539

3.1.2.2. Aldiwan Corpus

Aldiwan corpus contains 1040 single versed poems and 26625 two versed poems. Each poem belongs to 1 of the 10 existent eras and 1 of the 18 existent topics.

Table 4-3: Aldiwan Corpus Statistic according to eras.

Eras	Single-versed Poems	Two-versed Poems	Total
العصر الجاهلي	16	608	624
العصر الإسلامي	0	114	114
العصر العباسي	56	9735	9791

العصر الايوبي	8	197	205
العصر العثماني	3	1642	1645
المخضرمون	0	214	214
العصر الاموي	81	3336	3417
العصر الأندلسي	0	2072	2072
العصر المملوكي	18	3756	3774
العصر الحديث	858	4951	5809

Table 4-4: Aldiwan Corpus Statistic according to topics.

Topics	Single-versed Poems	Two-versed Poems	Total
قصيدة اعتذار	2	2	4
قصيدة حزينه	36	1082	1118
قصيدة دينية	8	235	243
قصيدة ذم	7	124	131
قصيدة رثاء	5	361	366
قصيدة رومنسيه	61	1774	1835
قصيدة عامه	736	7270	8006
قصيدة عتاب	17	1126	1143
قصيدة غزل	12	543	555
قصيدة شوق	14	451	465
قصيدة فراق	6	310	316
قصيدة قصيره	88	10760	10848
قصيدة مدح	12	2012	2024
قصيدة هجاء	23	478	501
قصيدة وطنيه	8	88	96
قصيدة سياسية	4	0	4
قصيدة المعلقات	0	6	6
قصيدة الاناشيد	1	3	4

3.2. Pre-Processing Phase

Document preprocessing, which is the first step in TC, converts the Arabic documents to a form that is suitable for classification tasks. These preprocessing tasks include a few linguistic tools such as tokenization, normalization, stop word removal, and stemming. These linguistic tools are used to reduce the ambiguity of words to increase the accuracy and effectiveness of the classification system.

The following figure 4-19 illustrates the Arabic pre-processing phase:

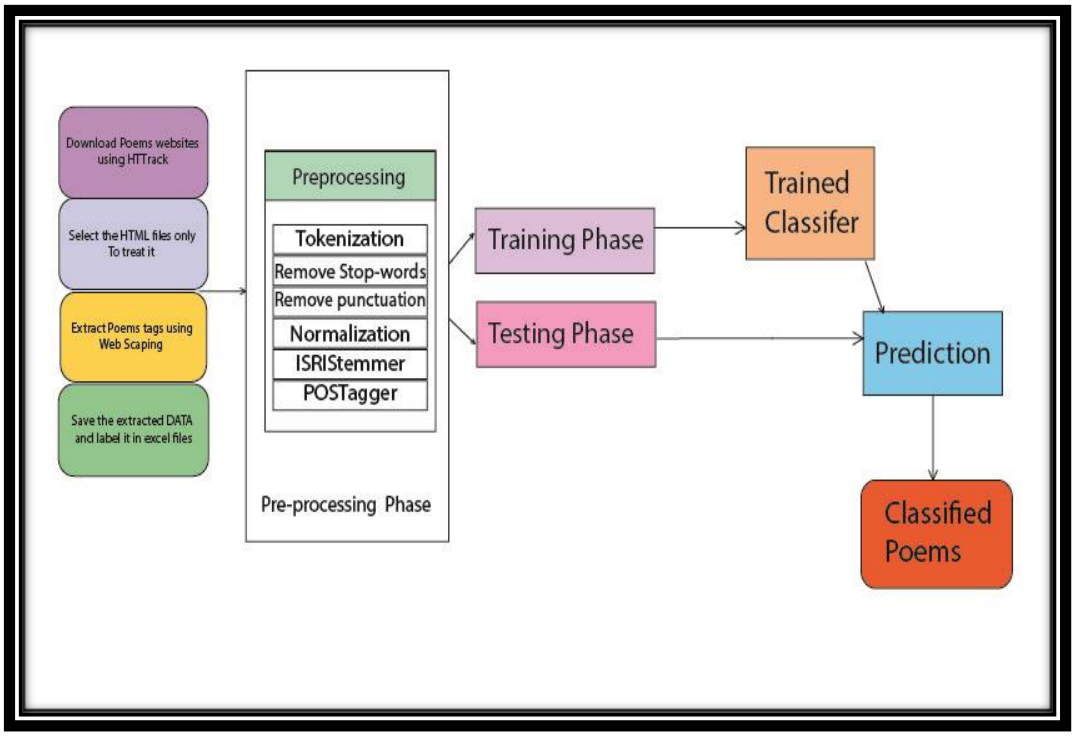


Figure 4-19: Arabic Language Pre-processing.

In pre-processing phase, we start with tokenization step to split the input into tokens, after that the stop words and punctuation removal steps. Then, the normalization and stemming steps and finally the post-tagging.

3.2.1. Tokenization

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms where a document is treated as a string, and then partitioned into a list of tokens.

Example:

Input:" أَغناهُ حُسنُ الجيدِ عَن لُبسِ الحُلي"

Output: "أَغناهُ","حُسنُ","الجيدِ","عَن","لُبسِ","الحُلي"

3.2.2. Stop-Words Removal

In the Arabic language, the stop-words are words having no significant semantic relation to the context in which they exist. They are extremely common terms that occur frequently in most of the documents.

Examples:

Input: "أَغناهُ حُسنُ الجيدِ عَن لُبسِ الحُلي "

Output: "أَغناهُ حُسنُ الجيدِ لُبسِ الحُلي "

Input: "أعلنت وزارة الصحة عن 138 اصابة بفيروس كورونا "

Output: "أعلنت وزارة الصحة 138 اصابة فيروس كورونا "

Input: "أعلن وزير الخارجية الأمريكي جون كيري أن الولايات المتحدة تنوي التعاون مع الباكستان "

Output: "أعلن وزير الخارجية الأمريكي جون كيري الولايات المتحدة تنوي التعاون الباكستان "

3.2.3. Punctuation Removal

Punctuation is a set of conventional signs for easing reading and understanding of the written text. The most commonly used group of characters are a comma (,), semicolon (;), two colon (:), full stop (.), exclamation mark (!), question mark (?), dash (-) and brackets, remove punctuation aims to delete all the punctuation marks.

Example:

Input: "قرَرتُ أن أشعُر سرًا: دونَما أن أستَعينَ بِفَمِي أو دَفتَري أو قَلَمي"

Output: "قرَرتُ أن أشعُر سرًا دونَما أن أستَعينَ بِفَمِي أو دَفتَري أو قَلَمي"

3.2.4. Normalization

Normalization aims to normalize certain letters that have different forms in the same word to one form. For example, the normalization of "ء" (hamza), "آ" (aleph mad), "أ" (aleph with hamza on top), "ؤ" (hamza on waw), "إ" (alef with hamza at the bottom), and "ئ" (hamza on ya) to "ا" (alef). Another example is the normalization of the letter "ى" to "ي" and the letter "ة" to "ه" [33].

3.2.5. ISRI Stemmer

In 2005, Kazem et al. have proposed the Information Science Research Institute's (ISRI) stemmer that shares many features with the Khoja stemmer. It uses a similar algorithm to word rooting of Khoja stemmer. However, it does not employ a root dictionary for lookup. In addition, if a word cannot be rooted, it is normalized by the ISRI stemmer (e.g. removing certain determinants and end patterns) instead of leaving the word unchanged. Furthermore, it defines sets of diacritical marks and affix classes. The ISRI stemmer has been shown to give good improvements to language tasks such as document clustering [33].

Example:

Term	→	ISRI
أفتضربونني	→	افتضربون

تستلزم	→	لزم
مكتبة	→	كتب
محامون	→	حام

3.2.6. POS Tagger

Part-of-speech tagging aims to assign parts of speech to each word of a given text (such as nouns, verbs, adjectives, and others) based on its definition and its context.

Example:

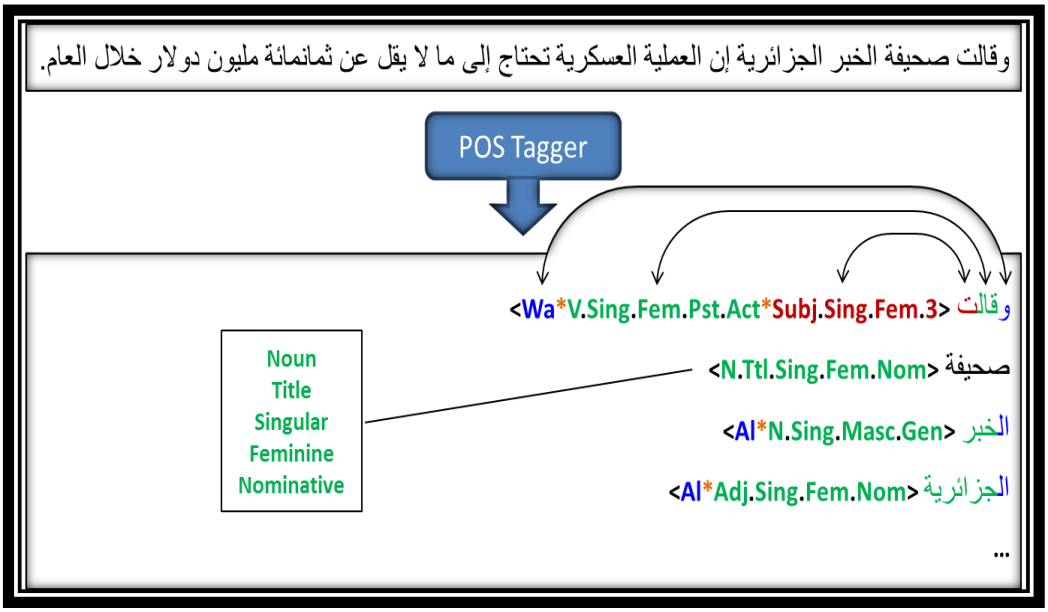


Figure 4-10: Part Of Speech Tagger Example.

3.3. Training Phase

A set of labeled poems, which is already prepared in the Preprocessing phase, is used as input to this phase. This phase is responsible for learning the classifier models. The output of this phase are trained classifiers which are ready for testing and classifying. It covers 80% of the original data.

In Adab corpus, we have 1988 poems in single versed poems and 1713 poems in two versed poems.

In Aldiwan corpus, we have 832 poems in single versed poems and 21300 poems in two versed poems.

The following figure 4-21 illustrates the training phase:

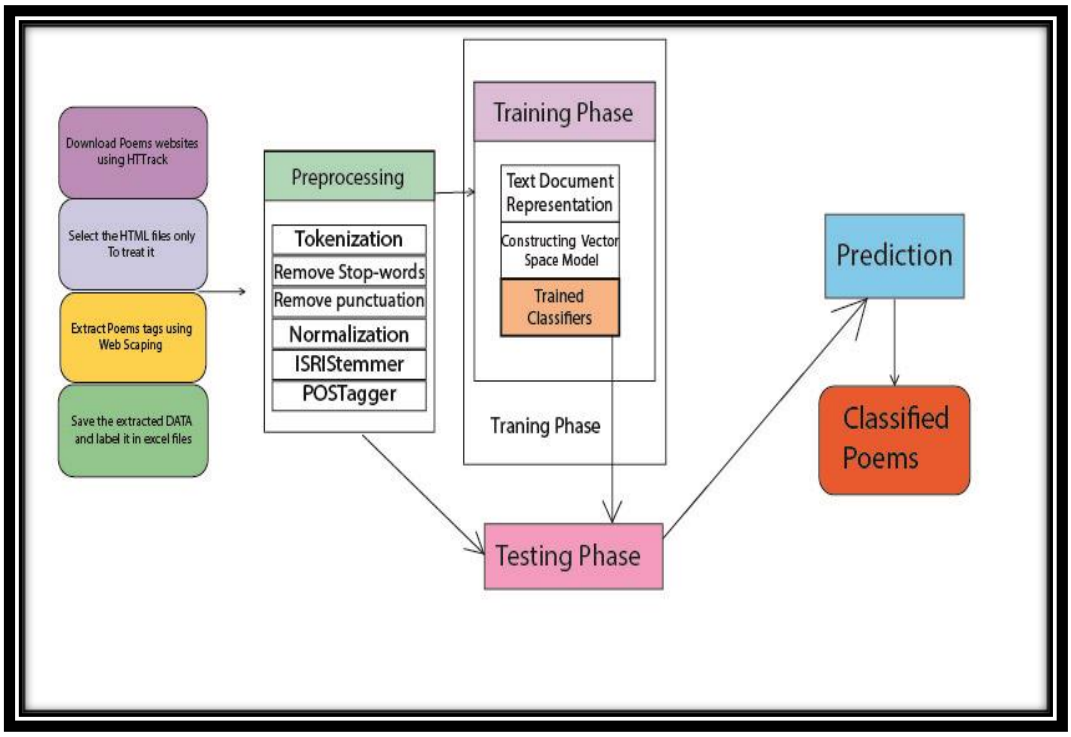


Figure 4-21: Training Phase.

After the pre-processing phase, the data is ready for the training phase. In this phase we represent the text document to a constructing vector space model and get as a result a trained classifiers.

4.3.1. Text Document Representation

Document representation is the process of presenting the words and their number of occurrences in each document. Our model uses the 5 gram for representing the text documents using the term frequency inverse document frequency.

4.3.2. Constructing Vector Space Model

We used Term Frequency Inverse Document Frequency weighting method to construct a vector space model to represent the features by using the previous method of text document representation (n-gram).

We used the parameter analyzer: {'word', 'char', 'char_wb'}, whether the feature should be made of word or character n-grams. Option 'char_wb' creates character n-grams only from text inside word boundaries; n-grams at the edges of words are padded with space.

TF-IDF can be calculated as:

$$W_{ij} = tf_{ij} \times idf_i = tf_{ij} \times \log\left(\frac{N}{df_i}\right)$$

Where W_{ij} is the weight of term i in document j, N is the number of all documents in the training set, tf_{ij} is the term frequency for term i in document j, df is the document frequency of term i in the documents of the training set.

4.3.3. Trained Classifiers

There is a lot of classification algorithms available now but it is not possible to conclude which one is superior to other. It depends on the application and nature of available dataset. The role of this phase is to build a classifier or generate model by training it using predefined documents that will be used to classify unlabeled documents.

In our works, we used 8 classifiers to do our experiments that we will see in the further chapter and selected the superior one that gave the best performance in the testing phase to apply it the prediction phase.

3.3.1.1. K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a lazy learning algorithm which stores all instances correspond to training data points in n-dimensional space. When an unknown

discrete data is received, it analyzes the closest k number of instances saved (nearest neighbors) and returns the most common class as the prediction and for real-valued data it returns the mean of k nearest neighbors.

KNN Pseudo Code

We can implement a KNN model by following the below steps [34]:

1. *Load the data;*
2. *Initialize the value of k;*
3. *For getting the predicted class, iterate from 1 to total number of training data points;*
4. *Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it is the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.*
5. *Sort the calculated distances in ascending order based on distance values;*
6. *Get top k rows from the sorted array;*
7. *Get the most frequent class of these rows;*
8. *Return the predicted class.*

3.3.1.2. Decision Tree

A Decision Tree (DT) is a Supervised Machine Learning algorithm which looks like an inverted tree, where in each node represents a predictor variable (feature), the link between the nodes represents a Decision and each leaf node represents an outcome (response variable) [35].

DT Pseudo Code

The Decision Tree Algorithm follows the below steps:

Step 1: Select the feature (predictor variable) that best classifies the data set into the desired classes and assign that feature to the root node.

Step 2: Traverse down from the root node, whilst making relevant decisions at each internal node such that each internal node best classifies the data.

Step 3: Route back to step 1 and repeat until you assign a class to the input data.

3.3.1.3. Linear Support Vector Classifier

In Linear Classifier, A data point consider as a p-dimensional vector (list of p-numbers) and we separate points using (p-1) dimensional hyperplane. There can be many hyperplanes separating data in a linear order, but the best hyperplane is consider to be the one which maximizes the margin i.e., the distance between hyperplane and closest data point of either class.

The Maximum-margin hyperplane is determine by the data points that lie nearest to it. Since we have to maximize the distance between hyperplane and the data points. These data points which influences our hyperplane are known as support vectors [36].

3.3.1.4. Logistic Regression

A logistic regression (LR) algorithm is a machine learning regression algorithm which measures the ways in which a set of data conforms to two particular variables. The algorithm dictates the variables, the relationship, and the ways in which the variables interact [37].

We use the Sigmoid function/curve to predict the categorical value. The threshold value decides the outcome(win/lose) [38]. Linear regression equation:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

- y stands for the dependent variable that needs to be predicted.

- β_0 is the Y-intercept, which is basically the point on the line which touches the y-axis.

- β_1 is the slope of the line (the slope can be negative or positive depending on the relationship between the dependent variable and the independent variable.)

- X here represents the independent variable that is used to predict our resultant dependent value.

Sigmoid function: $p = \frac{1}{1} + e^{-y}$

Apply sigmoid function on the linear regression equation.

Logistic Regression equation: $p = \frac{1}{1} + e^{-\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n}$

3.3.1.5. Multinomial Naïve Bayes

Multinomial Naive Bayes (MNB) algorithm is useful to model feature vectors where each value represents the number of occurrences of a term or its relative frequency. For example, if a feature vector has n elements and each of them can assume k different values with probability pk, then:

$$P(X_1 = x_1 \cap X_2 = x_2 \cap \ldots \cap X_k = x_k) = \frac{n!}{\prod_i x_i!} \prod_i {p_i}^{x_i}$$

The conditional probabilities P (xi | y) are computed with a frequency count. The frequency count corresponds to applying a maximum likelihood approach. During Multinomial Bayes Formula, Laplace smoothing factor is to be kept in mind. Its default value is 1.0 and prevents the model from setting null probabilities when the frequency is zero [39].

3.3.1.6. Bernoulli Naïve Bayes

Bernoulli Naive Bayes (BNB) is a part of the family of Naive Bayes. It only takes binary values. The most general example is where we check if each value will be whether or not a word that appears in a document. That is a very simplified model. In cases where counting the word frequency is less important, Bernoulli may give better results. In simple words, we have to count every value binary term occurrence

features i.e. a word occurs in a document or not. These features are used rather than finding the frequency of a word in the document [40].

3.3.1.7. Random Forest

Random forest (RF) algorithm is a supervised classification and regression algorithm. As the name suggests, this algorithm randomly creates a forest with several trees.

Generally, the more trees in the forest the more robust the forest looks like. Similarly, in the random forest classifier, the higher the number of trees in the forest, greater is the accuracy of the results.

In simple words, Random forest builds multiple decision trees (called the forest) and glues them together to get a more accurate and stable prediction. The forest it builds is a collection of Decision Trees, trained with the bagging method [41].

RF Pseudo Code

The Random Forest Algorithm follows the below steps:

Step 1: Create a Bootstrapped Data Set, Bootstrapping is an estimation method used to make predictions on a data set by re-sampling it. To create a bootstrapped data set, we must randomly select samples from the original data set. A point to note here is that we can select the same sample more than once.

Step 2: Creating Decision Trees, build a Decision Tree by using the bootstrapped data set created in the previous step. Since we are making a Random Forest, we will not consider the entire data set that we created, instead we will only use a random subset of variables at each step.

Step 3: Go back to Step 1 and Repeat, Random Forest is a collection of Decision Trees. Each Decision Tree predicts the output class based on the respective predictor variables used in that tree. Finally, the outcome of all the Decision Trees in a Random Forest is recorded and the class with the majority votes is computed as the output class.

Step 4: Predicting the outcome of a new data point. Bootstrapped the data and used the aggregate from all the trees to make a decision, this process is known as Bagging.

Step 5: Evaluate the Model.

3.3.1.8. Gradient Boosting

Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set.

The Gradient Boosting Algorithm can be most easily explained by first introducing the AdaBoost Algorithm. The AdaBoost Algorithm begins by training a decision tree in which each observation is assigned an equal weight. After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify. The second tree is therefore grown on this weighted data. Here, the idea is to improve upon the predictions of the first tree [42].

3.4. Testing Phase

This phase is responsible for testing the performance of the trained classifiers and evaluating their capability for the usage (prediction). The Main inputs of this phase are the trained classifiers from the previous phase and labeled testing poems. It covers 20% of the original data.

In Adab corpus, we have 498 poems in single versed poems and 429 poems in two versed poems.

In Aldiwan corpus, we have 208 poems in single versed poems and 5325 poems in two versed poems.

The following figure 4-22 illustrates the testing phase:

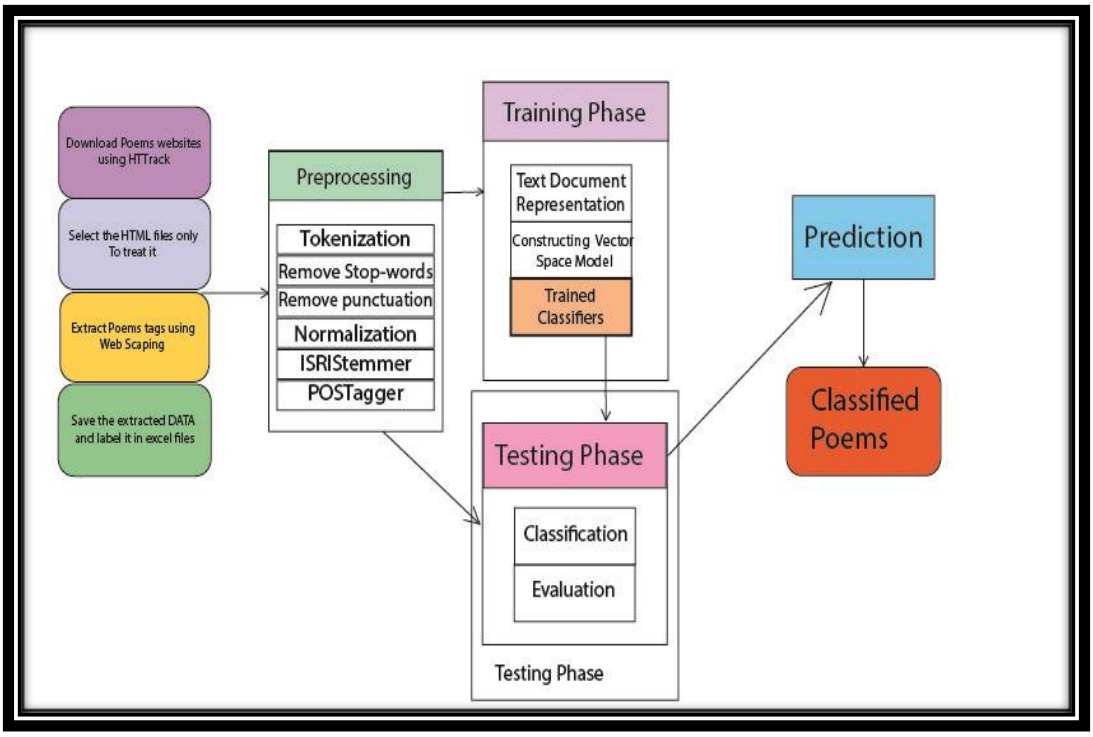


Figure 4-22: Testing Phase.

In this phase, we have trained classifiers from the previous phase (training phase) that we will applicate the classification step on it and evaluate them to finally choose the best model.

3.4.1. Classification

Categorizing a new poem by applying the trained classifiers generated in the learning phase to predict the class (Category or Era or Topic) of that poem. The model receives a poem as input and assigns a label (class) to it as output.

3.4.2. Evaluation

For evaluating the performance of those trained classifiers, the input is the predicted class labels of the testing documents from the classification step and the actual associated class labels. The performance of the classifiers are evaluated according to the results of comparing the predicted class labels with the actual labels.

After evaluation the performance of the eights classifiers, we found out that LSVC is the one with the higher accuracy; then it is the best for our datasets.

The following figure explain how the original data should be divided after the step of Pre-Processing.

Figure 4-23: Slicing a single data set into a training set and test set.

3.5. Prediction Phase

The more suitable classifier “LSVC” in this phase is successfully trained, tested and evaluated and ready for classification the new data. As a result, we had a categorized poems.

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data we provide, returning a "best fit" hyperplane that categorizes our data. From there, after getting the hyperplane, we can then feed some features to our classifier to see what the "predicted" class is.

The following figure 4-24 illustrates the prediction phase step and the classify poems step.

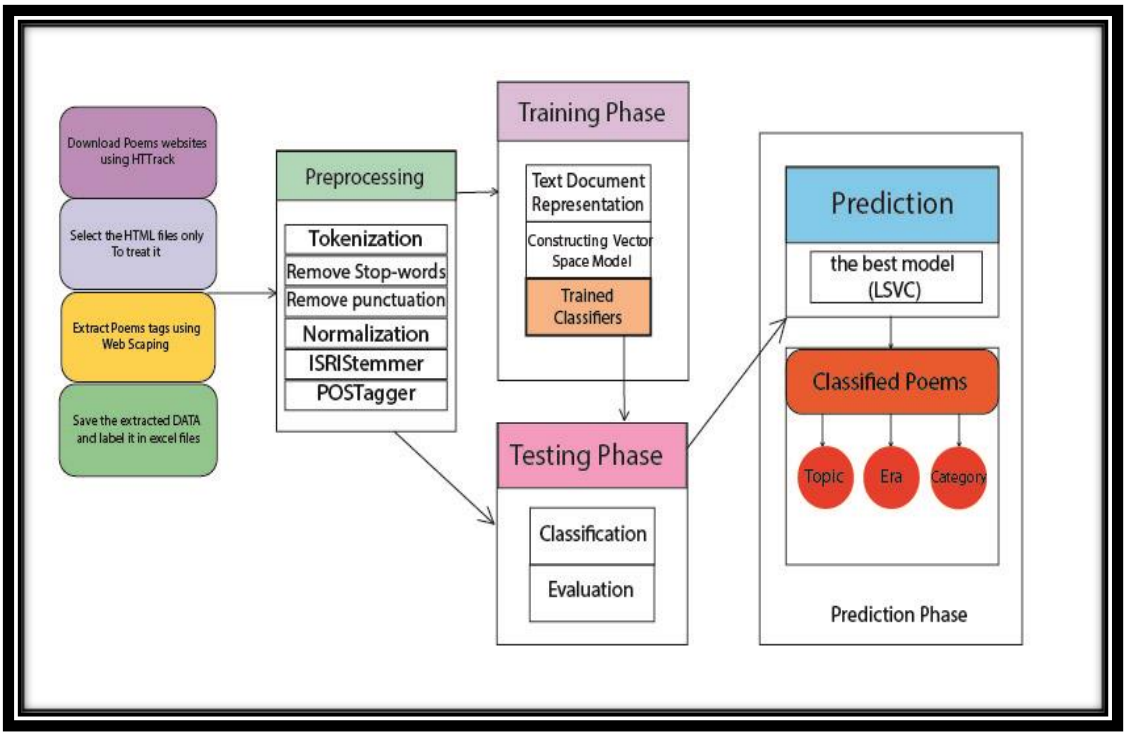


Figure 4-24: The Prediction Phase Step.

The final phase (prediction phase), from the best model we have; we try to predict the class of the given poems (from test data) according to their category, era and topic.

4. Conclusion

In this chapter, we presented the architecture of our proposed solution to classify Arabic poems according to their Eras and Topics, starting from corpora collecting, moving to the training phases with eight classifiers to be trained. Then, in testing phase we evaluated the eight trained classifiers and we finished by only one trained, tested and evaluated model that we used in the prediction step.

In the next chapter, we will see how we have done all the experiences and how we eliminated the 8 trained classifiers to only retain one. Finally, we present the results that we achieved along with the implementation of our application.

Chapter V: Experimental Results & Implementation

1. Introduction

In this chapter, we are going to present the implementation of our system "Saline Classification". First, we start by the presentation of the programming language, the development environments and tools that we used to build our classification system.

Second, we explain the experiments that we did to the both corpus that we created. Then, we analyze the results.

Finally, we explain the application under the name of "Saline Classification" that contains three services of prediction of poems according to category, era and topic.

2. Environments and Tools

In this section, we will talk about the python language and some of its libraries that we used in our work.

2.1. Language and Libraries

We use as language for coding "Python" which is a multiparadigm, general-purpose, object-oriented, interpreted, high-level programming language. It lets you work quickly and integrate systems more effectively.

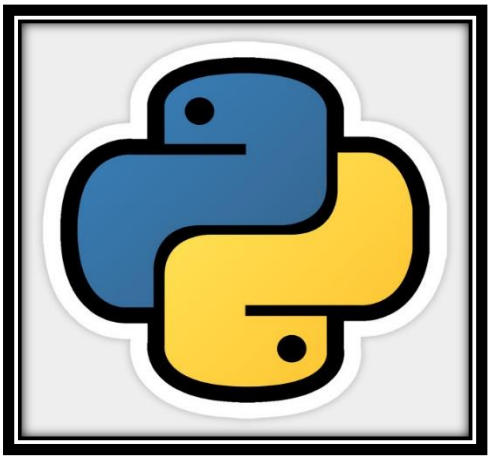


Figure 5-1: Python Logo.

Python is an ocean of libraries that serve various purposes like machine learning, in our case we utilize some of them:

- **NLTK library**

  NLTK is a leading platform for building Python programs to work with human language data.

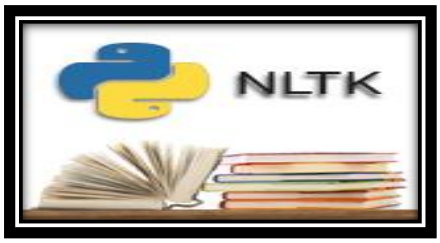


Figure 5-2: NLTK Logo.

We use it for some pre-processing operations like it is shown in the next figure:

```
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk import word_tokenize
from nltk.stem.isri import ISRIStemmer
from nltk.stem import WordNetLemmatizer
```

Figure 5-3: NLTK Code.

- **Scikit Learn**

  Scikit Leann Is a simple and effective tools for prediction data analysis.

Figure 5-4: Scikit Lean Logo.

This library was used for the classifiers, feature extraction and for evaluation measures as shown in Figures 5-5 and 5-6 present the code.

```
from sklearn.svm import LinearSVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import BernoulliNB, MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
```

Figure 5-5: Sklearn Classifiers.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import FeatureUnion
```

Figure 5-6: Sklearn Feature Code.

```
from sklearn.metrics import  classification_report
from sklearn.metrics import accuracy_score
```

Figure 5-7: Sklearn Metrics Code.

- **Pandas**

  Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool.

Figure 5-8: Pandas Logo.

Pandas was used for opening, reading and organizing our data set, the following figure present some operations:

```
def read_dataset(dataset, columnNameList):
  if dataset.split('.')[1]=='xlsx':
    df = pd.read_excel(dataset, encoding='utf-8')
  elif dataset.split('.')[1]=='csv':
    df = pd.read_csv(dataset, encoding='utf-8')
  text = df[columnNameList.split(', ')[0]].astype(str)
  label = df[columnNameList.split(', ')[1]].astype(str)
  return text, label
```

Figure 5-9: Pandas Code.

- **Pickle**

  Pickle implents a fundamental but powerfull algorithm for serialzing and de-serialzing a Python object structure.

Figure 5-10: Pickle Logo.

We use pickle library for loading files like it is shown in the following figure:

```
file = open(vecFileName, 'rb')
vectorizer = pickle.load(file)
matTrain = vectorizer.fit_transform(dataTrainText)
```

Figure 5-11: Pickle Code.

- **Re (Regular Expression)**

We used Re for Arabic normalization operation, figure 5-12 present the function:

```
def normalizeArabic(text):
    text = re.sub("[إأآا]" ,"[ا]", text)
    text = re.sub("ى" ,"ي", text)
    text = re.sub("ؤ" ,"ء", text)
    text = re.sub("ئ" ,"ء", text)
    return(text)
```

Figure 5-12: Re Code.

2.2. Environments

- **Spyder**

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. We use spyder for structuring data from the html files downloaded, and developing our classification system.

Figure 5-13: Syder Logo.

- **Google Colab**

Google Colab is a Jupyter notebook environment that runs completely on a cloud. It handles all the setup and configuration required for your program. It was used for executing the training commands.

Figure 5-14: Google Colab Logo.

- **Atom**

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub. We used it for developing the application.

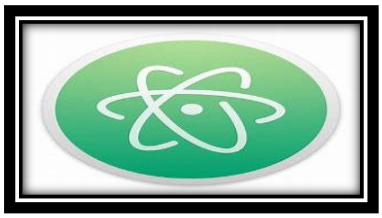


Figure 5-15: Atom Logo.

2.3. Tools

To build our corpora and application we use the following tools:

- **HTTrack**

HTTrack web site copier is a free and open-source web crawler and offline browser, developed by Xavier Roche and licensed under the GNU General Public License Version 3. HTTrack allows users to download World Wide Web sites from the Internet to a local computer. We used it for downloading the content of the web sites from where we collect our corpora.

Figure 5-16: HTTrack Logo.

- **Flask**

Flask is a Python web framework built with a small core and easy-to-extend philosophy. We used it for developing our application.

Figure 5-17: Flask Logo.

3. Preprocessing

In this section, we will present portions of codes of the pre-processing step.

3.1. Tokenization

From NLTK library we import word_tokenize to do the tokenization preprocessing.

3.2. Stop Words Removal

```
# filter arabic stopwords
def removeStopWord(text, lang):
  stop = stopwords.words(lang)
  out = text.apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
  return out
```

Figure 5-18: Stopwords Removal Function.

3.3. Punctuation Removal

```
# remove ponctuations
def removePunctuation(text):
  out = text.str.replace('[^\w\s]','')
  return out
```

Figure 5-19: Punctuation Removal Function.

3.4. Normalization

```
# normalize Arabic Letters
def textNormalizer(text):
  out = text.apply(lambda x: normalizeArabic(x))
  return out
```

Figure 5-20: Normalization Function.

3.5. Stemmer

```
# apply steeming
def applyStemmer(text):
  st = ISRIStemmer()
  out = ' '.join([st.stem(word) for word in text.split()])
  return out
```

Figure 5-21: Stemmer Function.

3.6. POS Tagger

```
# apply pos tagging
def applyPosTag(text):
  wpt = nltk.WordPunctTokenizer()
  # sentences 1
  out = []
  for s in text:
    text = wpt.tokenize(s)
    text_tagged = nltk.pos_tag(text)
    new_text = []
    new_tag = []
    for word in text_tagged:
        new_text.append(word[0])
        new_tag.append(word[1])
    s1 = new_text + new_tag
    out.append(' '.join(s1))
  return out
```

Figure 5-22: POS Tagger Function.

4. Experiments

To do our experiments, we created a command script to simplify our work and to give us all the experiments that we could do with two operations. In this script, we have 6 stages and each one have different numbers of experiments.

The two operations are:

- Analyzer

```
opt0 = ['-vec word', '-vec char', '-vec char_wb', '-vec all']
```

Figure 5-23: Analyzer Operator.

- Pre-processing

```
opt1 = ['-s "arabic"','-p True','-n True','-st True','-pos True']
```

Figure 5-24: Pre-processing Operator.

And the number of the experiments is the combination of opt0 and opt1 which is equal to 128.

Step 1

In this step, we worked with Adab corpus; we trained the classifiers according to their category with single versed poems.

- **Stage 0:** In this stage we have 4 experiments, we did not use any preprocessing configurations with the 8 classifiers like it is shown in the figure below:

Run	Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
1	KNN	-	Union5grams	-	0.740964	0.81
	LSVC	-	Union5grams	-	0.893574	0.9
	LR	-	Union5grams	-	0.837349	0.86
	DT	-	Union5grams	-	0.787149	0.79
	MNB	-	Union5grams	-	0.714859	0.83
	BNB	-	Union5grams	-	0.714859	0.83
	RF	-	Union5grams	-	0.783133	0.82
	GB	-	Union5grams	-	0.885542	0.89
2	KNN	-	Union5grams	-	0.783133	0.81
	LSVC	-	Union5grams	-	0.779116	0.82
	LR	-	Union5grams	-	0.714859	0.83
	DT	-	Union5grams	-	0.680723	0.68
	MNB	-	Union5grams	-	0.712851	0.83
	BNB	-	Union5grams	-	0.710843	0.83
	RF	-	Union5grams	-	0.74498	0.82
	GB	-	Union5grams	-	0.787149	0.81
3	KNN	-	Union5grams	-	0.730924	0.8
	LSVC	-	Union5grams	-	0.889558	0.9
	LR	-	Union5grams	-	0.829317	0.86
	DT	-	Union5grams	-	0.771084	0.78
	MNB	-	Union5grams	-	0.714859	0.83
	BNB	-	Union5grams	-	0.7249	0.82
	RF	-	Union5grams	-	0.805221	0.84
	GB	-	Union5grams	-	0.883534	0.89
4	KNN	-	Union5grams	-	0.716867	0.78
	LSVC	-	Union5grams	-	0.881526	0.89
	LR	-	Union5grams	-	0.811245	0.85
	DT	-	Union5grams	-	0.791165	0.79
	MNB	-	Union5grams	-	0.714859	0.83
	BNB	-	Union5grams	-	0.730924	0.82
	RF	-	Union5grams	-	0.799197	0.84
	GB	-	Union5grams	-	0.871486	0.88

Figure 5-25: Stage 0 Experiments Results.

- **Stage 1:** In this stage, we have 20 experiments and we used the 8 classifiers with the preprocessing configurations like it is shown in the figure below:

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
KNN	RemovePunct	Union5grams	-	0.740964	0.81
LSVC	RemovePunct	Union5grams	-	0.893574	0.9
LR	RemovePunct	Union5grams	-	0.837349	0.86
DT	RemovePunct	Union5grams	-	0.7751	0.77
MNB	RemovePunct	Union5grams	-	0.714859	0.83
BNB	RemovePunct	Union5grams	-	0.714859	0.83
RF	RemovePunct	Union5grams	-	0.761044	0.81
GB	RemovePunct	Union5grams	-	0.881526	0.89
KNN	Norm	Union5grams	-	0.740964	0.81
LSVC	Norm	Union5grams	-	0.893574	0.9
LR	Norm	Union5grams	-	0.837349	0.86
DT	Norm	Union5grams	-	0.781124	0.78
MNB	Norm	Union5grams	-	0.714859	0.83
BNB	Norm	Union5grams	-	0.714859	0.83
RF	Norm	Union5grams	-	0.767068	0.81
GB	Norm	Union5grams	-	0.881526	0.89

Figure 5-26: Stage 1 Experiments Results Part 1.

KNN	RemovePunct	Union5grams	-	0.783133	0.81
LSVC	RemovePunct	Union5grams	-	0.779116	0.82
LR	RemovePunct	Union5grams	-	0.714859	0.83
DT	RemovePunct	Union5grams	-	0.664659	0.65
MNB	RemovePunct	Union5grams	-	0.712851	0.83
BNB	RemovePunct	Union5grams	-	0.710843	0.83
RF	RemovePunct	Union5grams	-	0.740964	0.82
GB	RemovePunct	Union5grams	-	0.791165	0.81
KNN	Norm	Union5grams	-	0.783133	0.81
LSVC	Norm	Union5grams	-	0.779116	0.82
LR	Norm	Union5grams	-	0.714859	0.83
DT	Norm	Union5grams	-	0.664659	0.65
MNB	Norm	Union5grams	-	0.712851	0.83
BNB	Norm	Union5grams	-	0.710843	0.83
RF	Norm	Union5grams	-	0.732932	0.82
GB	Norm	Union5grams	-	0.787149	0.81
KNN	RemovePunct	Union5grams	-	0.730924	0.8
LSVC	RemovePunct	Union5grams	-	0.889558	0.9

Figure 5-27: Stage 1 Experiments Results Part 2.

LR	RemovePunct	Union5grams	-	0.829317	0.86
DT	RemovePunct	Union5grams	-	0.759036	0.76
MNB	RemovePunct	Union5grams	-	0.714859	0.83
BNB	RemovePunct	Union5grams	-	0.7249	0.82
RF	RemovePunct	Union5grams	-	0.787149	0.83
GB	RemovePunct	Union5grams	-	0.873494	0.88
KNN	Norm	Union5grams	-	0.730924	0.8
LSVC	Norm	Union5grams	-	0.889558	0.9
LR	Norm	Union5grams	-	0.829317	0.86
DT	Norm	Union5grams	-	0.769076	0.77
MNB	Norm	Union5grams	-	0.714859	0.83
BNB	Norm	Union5grams	-	0.7249	0.82
RF	Norm	Union5grams	-	0.787149	0.83
GB	Norm	Union5grams	-	0.873494	0.88
KNN	-	Union5grams	Stemmer	0.740964	0.81
LSVC	-	Union5grams	Stemmer	0.893574	0.9
LR	-	Union5grams	Stemmer	0.837349	0.86
DT	-	Union5grams	Stemmer	0.781124	0.78

Figure 5-28: Stage 1 Experiments Results Part 3.

MNB	-	Union5grams	Stemmer	0.714859	0.83
BNB	-	Union5grams	Stemmer	0.714859	0.83
RF	-	Union5grams	Stemmer	0.761044	0.81
GB	-	Union5grams	Stemmer	0.881526	0.89
KNN	-	Union5grams	Stemmer	0.783133	0.81
LSVC	-	Union5grams	Stemmer	0.779116	0.82
LR	-	Union5grams	Stemmer	0.714859	0.83
DT	-	Union5grams	Stemmer	0.670683	0.66
MNB	-	Union5grams	Stemmer	0.712851	0.83
BNB	-	Union5grams	Stemmer	0.710843	0.83
RF	-	Union5grams	Stemmer	0.732932	0.82
GB	-	Union5grams	Stemmer	0.787149	0.81
KNN	-	Union5grams	Stemmer	0.730924	0.8
LSVC	-	Union5grams	Stemmer	0.889558	0.9
LR	-	Union5grams	Stemmer	0.829317	0.86
DT	-	Union5grams	Stemmer	0.773092	0.78
MNB	-	Union5grams	Stemmer	0.714859	0.83
BNB	-	Union5grams	Stemmer	0.7249	0.82

Figure 5-29: Stage 1 Experiments Results Part 4.

RF	-	Union5grams	Stemmer	0.785141	0.83
GB	-	Union5grams	Stemmer	0.875502	0.88
KNN	-	Union5grams	PosTagger	0.740964	0.81
LSVC	-	Union5grams	PosTagger	0.893574	0.9
LR	-	Union5grams	PosTagger	0.837349	0.86
DT	-	Union5grams	PosTagger	0.785141	0.78
MNB	-	Union5grams	PosTagger	0.714859	0.83
BNB	-	Union5grams	PosTagger	0.714859	0.83
RF	-	Union5grams	PosTagger	0.777108	0.82
GB	-	Union5grams	PosTagger	0.885542	0.89
KNN	-	Union5grams	PosTagger	0.783133	0.81
LSVC	-	Union5grams	PosTagger	0.779116	0.82
LR	-	Union5grams	PosTagger	0.714859	0.83
DT	-	Union5grams	PosTagger	0.684739	0.68
MNB	-	Union5grams	PosTagger	0.712851	0.83
BNB	-	Union5grams	PosTagger	0.710843	0.83
RF	-	Union5grams	PosTagger	0.73494	0.82
GB	-	Union5grams	PosTagger	0.783133	0.8

Figure 5-30: Stage 1 Experiments Results Part 5.

KNN	-	Union5grams	PosTagger	0.730924	0.8
LSVC	-	Union5grams	PosTagger	0.889558	0.9
LR	-	Union5grams	PosTagger	0.829317	0.86
DT	-	Union5grams	PosTagger	0.75502	0.76
MNB	-	Union5grams	PosTagger	0.714859	0.83
BNB	-	Union5grams	PosTagger	0.7249	0.82
RF	-	Union5grams	PosTagger	0.795181	0.84
GB	-	Union5grams	PosTagger	0.879518	0.89
KNN	RemovePunct	Union5grams	-	0.716867	0.78
LSVC	RemovePunct	Union5grams	-	0.881526	0.86
LR	RemovePunct	Union5grams	-	0.811245	0.85
DT	RemovePunct	Union5grams	-	0.787149	0.79
MNB	RemovePunct	Union5grams	-	0.714859	0.83
BNB	RemovePunct	Union5grams	-	0.730924	0.82
RF	RemovePunct	Union5grams	-	0.807229	0.83
GB	RemovePunct	Union5grams	-	0.869478	0.87
KNN	Norm	Union5grams	-	0.716867	0.78
LSVC	Norm	Union5grams	-	0.881526	0.89

Figure 5-31: Stage 1 Experiments Results Part 6.

LR	Norm	Union5grams	-	0.811245	0.85
DT	Norm	Union5grams	-	0.787149	0.78
MNB	Norm	Union5grams	-	0.714859	0.83
BNB	Norm	Union5grams	-	0.730924	0.82
RF	Norm	Union5grams	-	0.807229	0.84
GB	Norm	Union5grams	-	0.869478	0.88
KNN	RemoveStopWords	Union5grams	-	0.74498	0.81
LSVC	RemoveStopWords	Union5grams	-	0.889558	0.89
LR	RemoveStopWords	Union5grams	-	0.829317	0.85
DT	RemoveStopWords	Union5grams	-	0.795181	0.79
MNB	RemoveStopWords	Union5grams	-	0.714859	0.83
BNB	RemoveStopWords	Union5grams	-	0.714859	0.83
RF	RemoveStopWords	Union5grams	-	0.779116	0.82
GB	RemoveStopWords	Union5grams	-	0.869478	0.88
KNN	RemoveStopWords	Union5grams	-	0.789157	0.82
LSVC	RemoveStopWords	Union5grams	-	0.76506	0.82
LR	RemoveStopWords	Union5grams	-	0.716867	0.83
DT	RemoveStopWords	Union5grams	-	0.662651	0.65

Figure 5-32: Stage 1 Experiments Results Part 7.

MNB	RemoveStopWords	Union5grams	-	0.712851	0.83
BNB	RemoveStopWords	Union5grams	-	0.710843	0.83
RF	RemoveStopWords	Union5grams	-	0.74498	0.82
GB	RemoveStopWords	Union5grams	-	0.785141	0.79
KNN	RemoveStopWords	Union5grams	-	0.738956	0.81
LSVC	RemoveStopWords	Union5grams	-	0.879518	0.89
LR	RemoveStopWords	Union5grams	-	0.825301	0.86
DT	RemoveStopWords	Union5grams	-	0.779116	0.78
MNB	RemoveStopWords	Union5grams	-	0.714859	0.83
BNB	RemoveStopWords	Union5grams	-	0.726908	0.82
RF	RemoveStopWords	Union5grams	-	0.789157	0.83
GB	RemoveStopWords	Union5grams	-	0.871486	0.88
KNN	-	Union5grams	Stemmer	0.716867	0.78
LSVC	-	Union5grams	Stemmer	0.881526	0.89
LR	-	Union5grams	Stemmer	0.811245	0.85
DT	-	Union5grams	Stemmer	0.793173	0.79
MNB	-	Union5grams	Stemmer	0.714859	0.83
BNB	-	Union5grams	Stemmer	0.730924	0.82

Figure 5-33: Stage 1 Experiments Results Part 8.

RF	-	Union5grams	Stemmer	0.799197	0.84
GB	-	Union5grams	Stemmer	0.871486	0.88
KNN	-	Union5grams	PosTagger	0.716867	0.78
LSVC	-	Union5grams	PosTagger	0.881526	0.89
LR	-	Union5grams	PosTagger	0.811245	0.85
DT	-	Union5grams	PosTagger	0.801205	0.8
MNB	-	Union5grams	PosTagger	0.714859	0.83
BNB	-	Union5grams	PosTagger	0.730924	0.82
RF	-	Union5grams	PosTagger	0.787149	0.83
GB	-	Union5grams	PosTagger	0.869478	0.88
KNN	RemoveStopWords	Union5grams	-	0.730924	0.8
LSVC	RemoveStopWords	Union5grams	-	0.881526	0.89
LR	RemoveStopWords	Union5grams	-	0.811245	0.85
DT	RemoveStopWords	Union5grams	-	0.799197	0.8
MNB	RemoveStopWords	Union5grams	-	0.714859	0.83
BNB	RemoveStopWords	Union5grams	-	0.728916	0.82
RF	RemoveStopWords	Union5grams	-	0.815261	0.85
GB	RemoveStopWords	Union5grams	-	0.869478	0.88

Figure 5-34: Stage 1 Experiments Results Part 9.

- **Stage 2:** In this stage, we have 40 experiments Then, we eliminate the last two classifiers according to their test score (accuracy) which are MNB and BNB and it leaves us with only 6 classifiers.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
KNN	RemovePunct+Norm	Union5grams	-	0.740964	0.81
LSVC	RemovePunct+Norm	Union5grams	-	0.893574	0.9
LR	RemovePunct+Norm	Union5grams	-	0.837349	0.86
DT	RemovePunct+Norm	Union5grams	-	0.779116	0.78
RF	RemovePunct+Norm	Union5grams	-	0.7751	0.82
GB	RemovePunct+Norm	Union5grams	-	0.881526	0.89
KNN	RemovePunct+Norm	Union5grams	-	0.783133	0.81
LSVC	RemovePunct+Norm	Union5grams	-	0.779116	0.82
LR	RemovePunct+Norm	Union5grams	-	0.714859	0.83
DT	RemovePunct+Norm	Union5grams	-	0.660643	0.65
RF	RemovePunct+Norm	Union5grams	-	0.730924	0.82
GB	RemovePunct+Norm	Union5grams	-	0.785141	0.8
KNN	Norm	Union5grams	Stemmer	0.740964	0.81
LSVC	Norm	Union5grams	Stemmer	0.893574	0.9
LR	Norm	Union5grams	Stemmer	0.837349	0.86
DT	Norm	Union5grams	Stemmer	0.785141	0.79

Figure 5-35: Stage 2 Experiments Results Part 1.

RF	Norm	Union5grams	Stemmer	0.761044	0.81
GB	Norm	Union5grams	Stemmer	0.883534	0.89
KNN	RemovePunct	Union5grams	Stemmer	0.783133	0.81
LSVC	RemovePunct	Union5grams	Stemmer	0.779116	0.82
LR	RemovePunct	Union5grams	Stemmer	0.714859	0.83
DT	RemovePunct	Union5grams	Stemmer	0.662651	0.65
RF	RemovePunct	Union5grams	Stemmer	0.728916	0.82
GB	RemovePunct	Union5grams	Stemmer	0.787149	0.81
KNN	Norm	Union5grams	Stemmer	0.783133	0.8
LSVC	Norm	Union5grams	Stemmer	0.779116	0.82
LR	Norm	Union5grams	Stemmer	0.714859	0.83
DT	Norm	Union5grams	Stemmer	0.674699	0.67
RF	Norm	Union5grams	Stemmer	0.736948	0.82
GB	Norm	Union5grams	Stemmer	0.787149	0.81
KNN	RemovePunct	Union5grams	Stemmer	0.730924	0.8
LSVC	RemovePunct	Union5grams	Stemmer	0.889558	0.9
LR	RemovePunct	Union5grams	Stemmer	0.829317	0.86
DT	RemovePunct	Union5grams	Stemmer	0.757028	0.76

Figure 5-36: Stage 2 Experiments Results Part 2.

RF	RemovePunct	Union5grams	Stemmer	0.791165	0.83
GB	RemovePunct	Union5grams	Stemmer	0.87751	0.88
KNN	Norm	Union5grams	PosTagger	0.740964	0.81
LSVC	Norm	Union5grams	PosTagger	0.893574	0.9
LR	Norm	Union5grams	PosTagger	0.837349	0.86
DT	Norm	Union5grams	PosTagger	0.787149	0.79
RF	Norm	Union5grams	PosTagger	0.777108	0.82
GB	Norm	Union5grams	PosTagger	0.881526	0.89
KNN	RemovePunct	Union5grams	PosTagger	0.783133	0.81
LSVC	RemovePunct	Union5grams	PosTagger	0.779116	0.82
LR	RemovePunct	Union5grams	PosTagger	0.714859	0.83
DT	RemovePunct	Union5grams	PosTagger	0.676707	0.67
RF	RemovePunct	Union5grams	PosTagger	0.730924	0.82
GB	RemovePunct	Union5grams	PosTagger	0.785141	0.8
KNN	Norm	Union5grams	PosTagger	0.783133	0.81
LSVC	Norm	Union5grams	PosTagger	0.779116	0.82
LR	Norm	Union5grams	PosTagger	0.714859	0.83
DT	Norm	Union5grams	PosTagger	0.674699	0.67

Figure 5-37: Stage 2 Experiments Results Part 3.

RF	Norm	Union5grams	PosTagger	0.736948	0.82
GB	Norm	Union5grams	PosTagger	0.787149	0.81
KNN	RemovePunct	Union5grams	PosTagger	0.730924	0.8
LSVC	RemovePunct	Union5grams	PosTagger	0.889558	0.9
LR	RemovePunct	Union5grams	PosTagger	0.829317	0.86
DT	RemovePunct	Union5grams	PosTagger	0.777108	0.78
RF	RemovePunct	Union5grams	PosTagger	0.789157	0.83
GB	RemovePunct	Union5grams	PosTagger	0.881526	0.89
KNN	-	Union5grams	PosTagger+Stemmer	0.783133	0.81
LSVC	-	Union5grams	PosTagger+Stemmer	0.779116	0.82
LR	-	Union5grams	PosTagger+Stemmer	0.714859	0.83
DT	-	Union5grams	PosTagger+Stemmer	0.676707	0.67
RF	-	Union5grams	PosTagger+Stemmer	0.73494	0.82
GB	-	Union5grams	PosTagger+Stemmer	0.785141	0.81
KNN	-	Union5grams	PosTagger+Stemmer	0.730924	0.8
LSVC	-	Union5grams	PosTagger+Stemmer	0.889558	0.9
LR	-	Union5grams	PosTagger+Stemmer	0.829317	0.86
DT	-	Union5grams	PosTagger+Stemmer	0.759036	0.76

Figure 5-38: Stage 2 Experiments Results Part 4.

RF	-	Union5grams	PosTagger+Stemmer	0.795181	0.84
GB	-	Union5grams	PosTagger+Stemmer	0.885542	0.89
KNN	RemoveStopWords+Norm	Union5grams	-	0.74498	0.81
LSVC	RemoveStopWords+Norm	Union5grams	-	0.889558	0.89
LR	RemoveStopWords+Norm	Union5grams	-	0.829317	0.85
DT	RemoveStopWords+Norm	Union5grams	-	0.803213	0.8
RF	RemoveStopWords+Norm	Union5grams	-	0.783133	0.83
GB	RemoveStopWords+Norm	Union5grams	-	0.865462	0.87
KNN	RemoveStopWords+RemovePunct	Union5grams	-	0.789157	0.82
LSVC	RemoveStopWords+RemovePunct	Union5grams	-	0.76506	0.82
LR	RemoveStopWords+RemovePunct	Union5grams	-	0.716867	0.83
DT	RemoveStopWords+RemovePunct	Union5grams	-	0.660643	0.64
RF	RemoveStopWords+RemovePunct	Union5grams	-	0.748996	0.82
GB	RemoveStopWords+RemovePunct	Union5grams	-	0.791165	0.82
KNN	RemoveStopWords+Norm	Union5grams	-	0.789157	0.82
LSVC	RemoveStopWords+Norm	Union5grams	-	0.76506	0.83
LR	RemoveStopWords+Norm	Union5grams	-	0.716867	0.66
DT	RemoveStopWords+Norm	Union5grams	-	0.748996	0.81

Figure 5-39: Stage 2 Experiments Results Part 5.

RF	RemoveStopWords+Norm	Union5grams	-	0.7751	0.78
GB	RemoveStopWords+Norm	Union5grams	-	0.869478	0.88
KNN	RemoveStopWords+RemovePunct	Union5grams	-	0.738956	0.81
LSVC	RemoveStopWords+RemovePunct	Union5grams	-	0.879518	0.89
LR	RemoveStopWords+RemovePunct	Union5grams	-	0.825301	0.86
DT	RemoveStopWords+RemovePunct	Union5grams	-	0.785141	0.79
RF	RemoveStopWords+RemovePunct	Union5grams	-	0.801205	0.84
GB	RemoveStopWords+RemovePunct	Union5grams	-	0.871486	0.88
KNN	RemoveStopWords	Union5grams	Stemmer	0.789157	0.82
LSVC	RemoveStopWords	Union5grams	Stemmer	0.76506	0.82
LR	RemoveStopWords	Union5grams	Stemmer	0.716867	0.83
DT	RemoveStopWords	Union5grams	Stemmer	0.656627	0.64
RF	RemoveStopWords	Union5grams	Stemmer	0.757028	0.82
GB	RemoveStopWords	Union5grams	Stemmer	0.789157	0.8
KNN	RemoveStopWords	Union5grams	Stemmer	0.738956	0.78
LSVC	RemoveStopWords	Union5grams	Stemmer	0.879518	0.89
LR	RemoveStopWords	Union5grams	Stemmer	0.825301	0.86
DT	RemoveStopWords	Union5grams	Stemmer	0.791165	0.79

Figure 5-40: Stage 2 Experiments Results Part 6.

RF	RemoveStopWords	Union5grams	Stemmer	0.789157	0.83
GB	RemoveStopWords	Union5grams	Stemmer	0.871486	0.88
KNN	RemoveStopWords	Union5grams	PosTagger	0.789157	0.82
LSVC	RemoveStopWords	Union5grams	PosTagger	0.76506	0.82
LR	RemoveStopWords	Union5grams	PosTagger	0.716867	0.83
DT	RemoveStopWords	Union5grams	PosTagger	0.716867	0.67
RF	RemoveStopWords	Union5grams	PosTagger	0.678715	0.82
GB	RemoveStopWords	Union5grams	PosTagger	0.789157	0.8
KNN	RemoveStopWords	Union5grams	PosTagger	0.738956	0.81
LSVC	RemoveStopWords	Union5grams	PosTagger	0.879518	0.89
LR	RemoveStopWords	Union5grams	PosTagger	0.825301	0.86
DT	RemoveStopWords	Union5grams	PosTagger	0.777108	0.77
RF	RemoveStopWords	Union5grams	PosTagger	0.783133	0.83
GB	RemoveStopWords	Union5grams	PosTagger	0.869478	0.88

Figure 5-41: Stage 2 Experiments Results Part 7.

- **Stage 3:** In this stage, we have also 40 experiments. Then, we eliminate the last two classifiers according to their test score (accuracy) which are KNN and DT and it leaves us with only 4 classifiers.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemovePunct+Norm	Union5grams	Stemmer	0.893574	0.9
LR	RemovePunct+Norm	Union5grams	Stemmer	0.85743	0.87
RF	RemovePunct+Norm	Union5grams	Stemmer	0.799197	0.84
GB	RemovePunct+Norm	Union5grams	Stemmer	0.883534	0.89
LSVC	RemovePunct+Norm	Union5grams	Stemmer	0.889558	0.9
LR	RemovePunct+Norm	Union5grams	Stemmer	0.829317	0.86
RF	RemovePunct+Norm	Union5grams	Stemmer	0.801205	0.84
GB	RemovePunct+Norm	Union5grams	Stemmer	0.879518	0.89
LSVC	RemovePunct+Norm	Union5grams	PosTagger	0.893574	0.9
LR	RemovePunct+Norm	Union5grams	PosTagger	0.85743	0.87
RF	RemovePunct+Norm	Union5grams	PosTagger	0.793173	0.83
GB	RemovePunct+Norm	Union5grams	PosTagger	0.883534	0.89
LSVC	RemovePunct	Union5grams	Stemmer+PosTagger	0.893574	0.9
LR	RemovePunct	Union5grams	Stemmer+PosTagger	0.85743	0.87
RF	RemovePunct	Union5grams	Stemmer+PosTagger	0.801205	0.84
GB	RemovePunct	Union5grams	Stemmer+PosTagger	0.881526	0.89

Figure 5-42: Stage 3 Experiments Results Part 1.

LSVC	Norm	Union5grams	Stemmer+PosTagger	0.893574	0.9
LR	Norm	Union5grams	Stemmer+PosTagger	0.85743	0.87
RF	Norm	Union5grams	Stemmer+PosTagger	0.807229	0.84
GB	Norm	Union5grams	Stemmer+PosTagger	0.879518	0.89
LSVC	RemovePunct	Union5grams	Stemmer+PosTagger	0.889558	0.9
LR	RemovePunct	Union5grams	Stemmer+PosTagger	0.829317	0.86
RF	RemovePunct	Union5grams	Stemmer+PosTagger	0.791165	0.83
GB	RemovePunct	Union5grams	Stemmer+PosTagger	0.879518	0.89
LSVC	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.76506	0.82
LR	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.716867	0.83
RF	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.751004	0.82
GB	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.783133	0.79
LSVC	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.879518	0.89
LR	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.825301	0.86
RF	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.791165	0.83
GB	emoveStopWords+RemovePunct+Norı	Union5grams	-	0.86747	0.87
LSVC	RemovePunct+Norm	Union5grams	Stemmer	0.881526	0.89

Figure 5-43: Stage 3 Experiments Results Part 2.

LR	RemovePunct+Norm	Union5grams	Stemmer	0.811245	0.85
RF	RemovePunct+Norm	Union5grams	Stemmer	0.799197	0.84
GB	RemovePunct+Norm	Union5grams	Stemmer	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.889558	0.89
LR	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.84739	0.86
RF	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.795181	0.83
GB	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.86747	0.87
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer	0.889558	0.89
LR	RemoveStopWords+Norm	Union5grams	Stemmer	0.84739	0.86
RF	RemoveStopWords+Norm	Union5grams	Stemmer	0.801205	0.84
GB	RemoveStopWords+Norm	Union5grams	Stemmer	0.865462	0.87
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.76506	0.82
LR	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.716867	0.83
RF	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.751004	0.82
GB	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.785141	0.79
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer	0.76506	0.82
LR	RemoveStopWords+Norm	Union5grams	Stemmer	0.716867	0.83

Figure 5-44: Stage 3 Experiments Results Part 3.

RF	RemoveStopWords+Norm	Union5grams	Stemmer	0.753012	0.82
GB	RemoveStopWords+Norm	Union5grams	Stemmer	0.785141	0.79
LSVC	RemovePunct+Norm	Union5grams	PosTagger	0.881526	0.89
LR	RemovePunct+Norm	Union5grams	PosTagger	0.761044	0.83
RF	RemovePunct+Norm	Union5grams	PosTagger	0.769076	0.82
GB	RemovePunct+Norm	Union5grams	PosTagger	0.86747	0.87
LSVC	RemovePunct	Union5grams	Stemmer+PosTagger	0.881526	0.89
LR	RemovePunct	Union5grams	Stemmer+PosTagger	0.761044	0.83
RF	RemovePunct	Union5grams	Stemmer+PosTagger	0.767068	0.82
GB	RemovePunct	Union5grams	Stemmer+PosTagger	0.871486	0.88
LSVC	Norm	Union5grams	Stemmer+PosTagger	0.881526	0.89
LR	Norm	Union5grams	Stemmer+PosTagger	0.761044	0.83
RF	Norm	Union5grams	Stemmer+PosTagger	0.769076	0.82
GB	Norm	Union5grams	Stemmer+PosTagger	0.871486	0.88
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	-	0.881526	0.89
LR	RemoveStopWords+RemovePunct+Norm	Union5grams	-	0.773092	0.83
RF	RemoveStopWords+RemovePunct+Norm	Union5grams	-	0.781124	0.82

Figure 5-45: Stage 3 Experiments Results Part 4.

GB	RemoveStopWords+RemovePunct+Norm	Union5grams	-	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.881526	0.89
LR	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.773092	0.83
RF	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.7751	0.82
GB	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.881526	0.89
LR	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.773092	0.83
RF	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.781124	0.82
GB	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.869478	0.88
LSVC	RemoveStopWords	Union5grams	Stemmer+PosTagger	0.881526	0.89
LR	RemoveStopWords	Union5grams	Stemmer+PosTagger	0.773092	0.83
RF	RemoveStopWords	Union5grams	Stemmer+PosTagger	0.7751	0.82
GB	RemoveStopWords	Union5grams	Stemmer+PosTagger	0.869478	0.88

Figure 5-46: Stage 3 Experiments Results Part 5.

- **Stage 4**: In this stage, we have 20 experiments. Then, we eliminate the last two classifiers according to their test score (accuracy) which are LR and RF which leave us with only 2 classifiers.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.893574	0.9
GB	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.881526	0.89
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.779116	0.82
GB	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.783133	0.8
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.889558	0.9
GB	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.883534	0.89
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.889558	0.89
GB	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.86747	0.87
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.76506	0.82
GB	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.785141	0.79
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.879518	0.89
GB	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.871486	0.88
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.889558	0.89
GB	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.76506	0.82

Figure 5-47: Stage 4 Experiments Results Part 1.

LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.76506	0.82
GB	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.781124	0.79
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.879518	0.89
GB	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.869478	0.88
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.881526	0.89
GB	RemovePunct+Norm	Union5grams	Stemmer+PostTagger	0.86747	0.87
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.889558	0.89
GB	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.86747	0.87
LSVC	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.889558	0.89
GB	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.76506	0.82
GB	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.785141	0.79
LSVC	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.76506	0.82
GB	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.797189	0.81
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.879518	0.89
GB	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.871486	0.88

Figure 5-48: Stage 4 Experiments Results Part 2.

LSVC	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.881526	0.89
GB	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.881526	0.89
GB	RemoveStopWords+RemovePunct	Union5grams	Stemmer	0.869478	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.881526	0.89
GB	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.871486	0.88
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.881526	0.89
GB	RemoveStopWords+RemovePunct	Union5grams	PosTagger+Stemmer	0.869478	0.88
LSVC	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.881526	0.89
GB	RemoveStopWords+Norm	Union5grams	PosTagger+Stemmer	0.869478	0.88

Figure 5-49: Stage 4 Experiments Results Part 3.

- **Stage 5:** In this Stage, we have 4 experiments. Then, we eliminate the last classifier according to its test score (accuracy) which is GB and it leaves us with the best classifier "LSVC".

Run	Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
1	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.889558	0.89
2	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.76506	0.82
3	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.879518	0.89
4	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.881526	0.89

Figure 5-50: Stage 5 Experiments Results.

Step 2

Choosing the 18 best commands: we take all the experiments in stage 5 and choose 14 experiments from stage 4 with test score more than 88.25% according to LSVC classifier. Then, we used it to train our model "LSVC" according to category with two-versed poems in Adab corpus.

Run	Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
1	LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.944916	0.95
2	LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.943218	0.94
3	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.943824	0.94
4	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.940791	0.94
5	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.943218	0.94
6	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.941276	0.94
7	LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.940791	0.94
8	LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.943218	0.94
9	LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.940791	0.94
10	LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.943582	0.94
11	LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.943218	0.94
12	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.943218	0.94
13	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.940791	0.94
14	LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.940791	0.94
15	LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.941276	0.94
16	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.941276	0.94
17	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.941276	0.94
18	LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.941276	0.94

Figure 5-51: Experiments Results for Two-Versed Poems With Adab Corpus according to Category.

And according to era for two-versed poems in corpus Adab because in single-versed poems, all the poems have the same era.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.872745	0.9
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.873558	0.9
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.874533	0.91
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.875833	0.91
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.873558	0.9
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.875833	0.91
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.867869	0.89
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.875833	0.91
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.873558	0.9
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.875833	0.91
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.868519	0.89
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.873558	0.9
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.873558	0.9
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.875833	0.91
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.875833	0.91
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.867869	0.89
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.867869	0.89
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.867869	0.89

Figure 5-52: Experiments Results For Two-Versed Poems With Adab Corpus according to Era.

Step 3

We choose 8 best commands to train our model according to era single-versed and two-versed poems for Aldiwan corpus. The figures below show the results:

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.946913	0.96
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.945889	0.96
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.946229	0.96
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.946229	0.96
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemme+PosTagger	0.945889	0.96
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.946229	0.96
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.944513	0.96
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.944513	0.96

Figure 5-53: Experiments Results For Single-Versed Poems With Aldiwan Corpus according to Era.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemovePunct+Norm	Union5grams	Stemmer+PosTagger	0.613659	0.62
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.613213	0.62
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.608819	0.62
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	PosTagger	0.608819	0.62
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemme+PosTagger	0.613213	0.62
LSVC	RemoveStopWords+RemovePunct	Union5grams	PosTagger	0.608819	0.62
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.604728	0.61
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.604728	0.61

Figure 5-54: Experiments Results For Two-Versed Poems With Aldiwan Corpus according to Era.

Step 4

We choose 4 best commands to train our model according to topic single-versed and two-versed poems for aldiwan corpus.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.764078	0.81
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.76627	0.81
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.764312	0.81
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.76627	0.81

Figure 5-55: Experiments Results For Single-Versed Poems With Aldiwan Corpus according to Topic.

Model	Text PreProcessing	TFIDF Features	Morpho Features	Test Score	F1-Score
LSVC	RemoveStopWords+Norm	Union5grams	Stemmer+PosTagger	0.385764	0.46
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.382062	0.45
LSVC	RemoveStopWords+RemovePunct+Norm	Union5grams	Stemmer	0.383964	0.45
LSVC	RemoveStopWords+RemovePunct	Union5grams	Stemmer+PosTagger	0.382062	0.45

Figure 5-56: Experiments Results For Two-Versed Poems With Aldiwan Corpus according to Topic.

Step 5

With these 4 experiments, we build our application “Saline classification” to predict our poems with LSVC classifier.

5. Evaluation

We used macro avg, weighted avg, recall, precision, accuracy and f1 score to evaluate the performance of our experiments. Then, we took f1 score and accuracy in consideration to eliminate the worst classifiers.

6. Results

Our approach consisted in the first place, on making comparisons with different configurations between the eight classifiers mentioned above in terms of f-score and test score (accuracy).

On the second place, after testing and evaluating each classifier; we saw that LSVC gave a better test score than the other seven algorithms.

Moreover, LSVC classifier gave a great result during the first corpus, when we did the classification according to category for the single-versed poems (88.95%) and for the two-versed poems (94.94%) and according to era for the two-versed poems (87.58%) and for the single-versed poems we did not do it because all the poems have the same era (العصر الحديث).

During the tests with the second corpus, our model gave also a great results when we did the classification according to era for the single-versed poems (94.69%) and it decrease for the two-versed poems to (61.36%) because of the size of our corpus and the number of classes (10 classes). And according to topic for the single-versed poems gave a great result (76.62%) and gave a good results for the two-versed poems (38.57%) because of the size of our corpus and number of classes (18 classes).

Therefore, we will choose the model generated by the LSVC classifier with 4 configurations during the implementation phase in order to establish the predictions.

7. Application

We have developed a classification prediction application called “Saline Classification”. It has several pages such us: Home page, Service page, About us page, Category page, Era page and Topic page for both single-versed and two-versed poems.

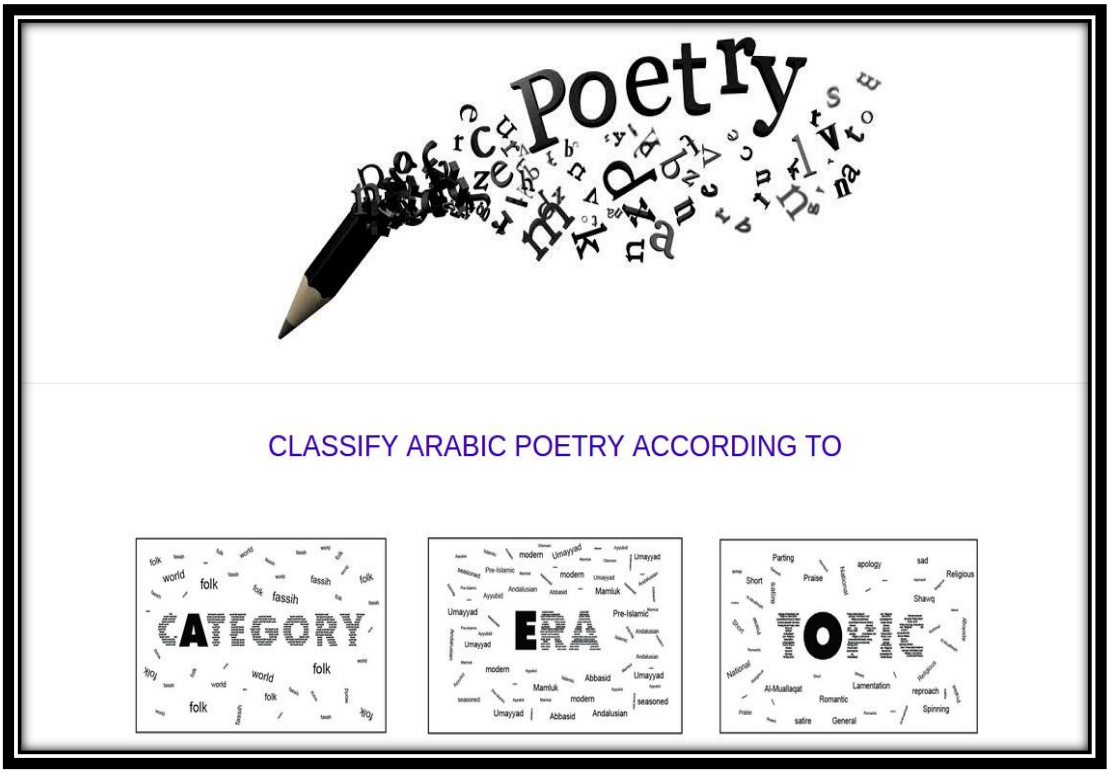


Figure 5-57: Service Page.

In this page:

- We can choose our service to classify our poetry according to category, era and topic.
- We can choose the type of our poem either single-versed or two-versed poems.

The following figures show Category pages.

Figure 5-58: Category Page For Single-Versed Poems.

Figure 5-59: Category Page For Two-Versed Poems.

In These pages:

- We can generate a random poem from Adab corpus and predict its category by, clicking on Classify button.

The following figures show Era pages.

Figure 5-60: Era Page For Single-Versed Poems.

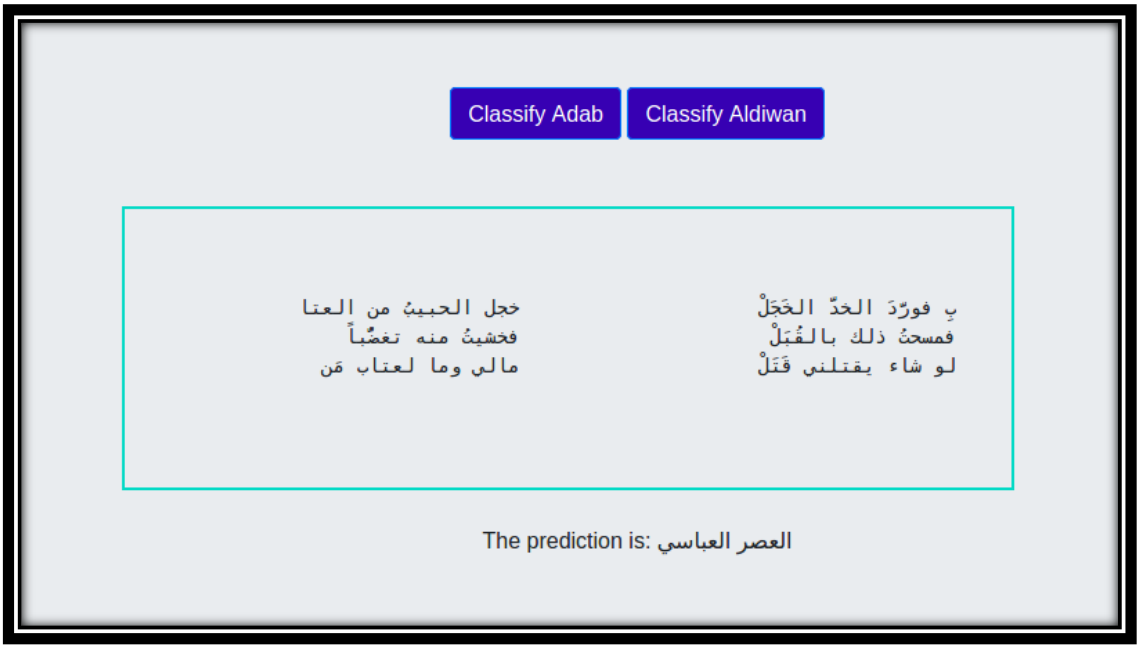


Figure 5-61: Era Page For Two-Versed Poems.

In These pages:

- We can generate a random poem from Adab corpus and predict its era by clicking on Classify Adab button
- We can generate a random poem from Aldiwan corpus and predict its era by clicking on Classify Aldiwan button.

The following figures show Topic pages.

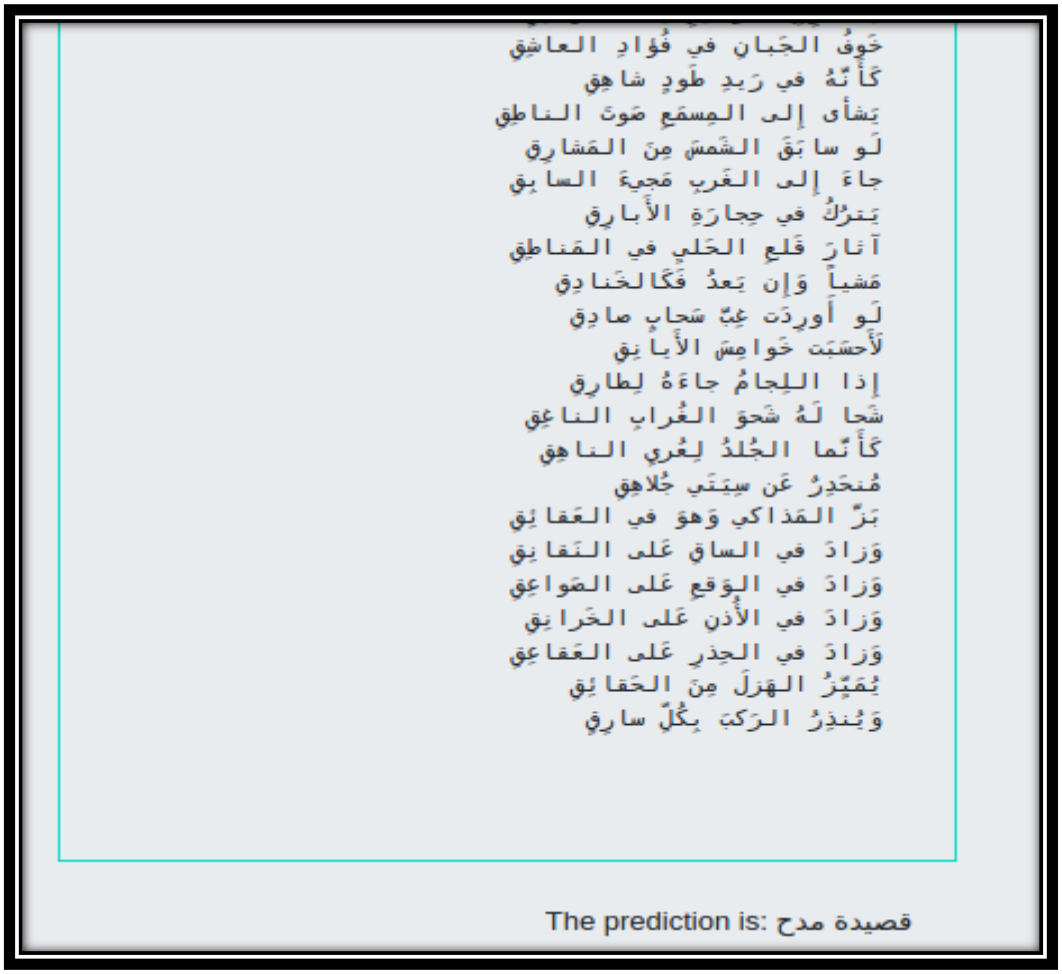


Figure 5-62: Topic Page For Single-Versed Poems.

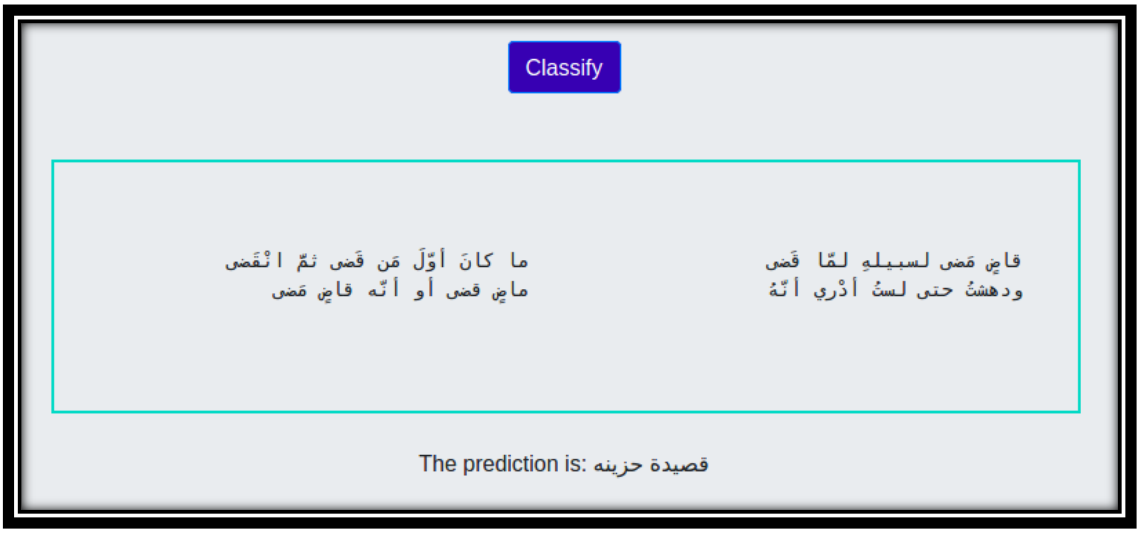


Figure 5-63: Topic Page For Two-Versed Poems.

In These pages:

- We can generate a random poem from Aldiwan corpus and predict its topic by clicking on Classify button.

8. Conclusion

The experiments allowed us to go through the results and evaluation of the Arabic Poems Classification, observe the problems and solve them.

The experiments showed us that pre-processing sometimes improves the accuracy and sometimes it decreases it because the structure of poetry can be very hard to handle. For example stop words can affect the poetry meaning and the same goes for the punctuation.

There is another issue which is the compatibility between training and testing set. The closer they are to each other, the better the prediction of the classification is, and vice versa.

Finally, we are convinced that further improvements can be made in this regard, which we expect to explore in future work.

General Conclusion

1. Conclusion

Poetry is abundant with emotions, despite being succinct or terse. The level of ambiguity and subtlety is remarkable. The complexity of computational study of poetry is the prime motivation for this dissertation. It important for not only can it lead to a better understanding of what makes rich literature, but it also has applications such as making recommendations to readers based on their literary tastes.

Our work consists on proposition of a classification system. This system is design for Arabic poetry to classify poems according to their eras and their topics. In the beginning of this work we have found difficulties that we fixed by creating corpora, analyze and made experiments. We hope that the results obtained are pretty good in comparison with the previous works even though the Arabic poetry is very challenging structure.

Our work consists of the classification of Arabic Poetry. To do so, we create two corpora:"Adab" & "Aldiwan", with the aim of classifying them according to category, era and topic.

- Adab corpus contains 2486 single versed poems and 2163 two versed poems.
- Aldiwan corpus contains 1040 single versed poems and 26625 two versed poems.

After collecting the corpora we done several experiments on it; we trained, tested and evaluated many classifiers (like we explained in the previous chapters) to give us in the end the best model that we applied it on our application.

Thanks to this work, we have now a good knowledge about the Arabic poetry thought it is very hard to handle and to understand it. We have also get a comprehension about the difficulties of the Arabic poetry and how to solve them.

Saline Classification project addresses the following aspects of poetry classification:

1. Category Classification.
2. Era Classification.
3. Topic Classification.

In category classification, we have one corpus “Adab” to predict from it the poem’s category which we have 3 categories.

In era classification, we have two corpora to predict from them the poem’s era which we have 4 eras in Adab corpus and 10 eras in Aldiwan corpus.

In category classification, we have one corpus “Aldiwan” to predict from it the poem’s topic which we have 18 topics.

2. Future Work

With the advent and percolation of deep learning into the field of NLP, it would be a perhaps naive thing to say that computational poetry and digital humanities as a whole will not be influenced by this advancement. We were quite encouraged by the results that we obtained and we hope to continue working in this direction in the future.

- Arabic poems classification according to seas (buhur).
- Poet Identification.
- Multi-output.

References

[1] R. H. Robins and D. Crystal, "language | Definition, Characteristics, & Change | Britannica," [Online]. Available: https://www.britannica.com/topic/language. [Accessed 2020].

[2] A. Farghaly and . K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Transactions on Asian Language Information Processing,* vol. 8, no. 4, 2009.

[3] K. Shaalan, "Rule-based Approach in Arabic Natural Language Processing Machine Translation View project Further Investigations on Developing an Arabic Sentiment Lexicon View project," International Journal on Information and Communication Technologies, 2010.

[4] S. O. E. ALAOUI, "Approches statistique et sémantique pour l'accès à l'information dans les collections textuelles en langue arabe," 2015.

[5] D. L. N. Harsa, "Introduction to Words and Morphemes," in *English Morpho-Syntax*, 2014.

[6] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni and D. Nouvel, "Arabic natural language processing: An overview," *Journal of King Saud University - Computer and Information Sciences,* 2019.

[7] E. Othman and A. Al-Hamadi, "Automatic Arabic Document Classification Based on the HRWiTD Algorithm," *Journal of Software Engineering and Applications,* vol. 11, no. 04, pp. 167-179, 2018.

[8] V. Korde, "Text Classification and Classifiers:A Survey," *International Journal of Artificial Intelligence & Applications,* vol. 2, no. 3, pp. 85-99, 2012.

[9] M. Ikonomakis, S. Kotsiantis and V. Tampakas, "Text classification using machine learning techniques," *WSEAS Transactions on Computers,* vol. 4, no. 8, pp. 966-974, 2005.

[10] F. Sebastiani, "Machine Learning in Automated Text Categorization," 2002.

[11] S. Gupta, "Automated Text Classification Using Machine Learning," ITNEXT, 2018. [Online]. Available: https://itnext.io/automated-text-classification-using-machine-learning-98dbe7f5e133. [Accessed 2020].

[12] M. Hassler and G. Fliedl, "Text Preparation Through Extended Tokenization," 2006.

[13] A. ZEGGADA and R. MOULAI, "CATEGORISATION AUTOMATIQUE DES TEXTES ARABES," 2019.

[14] A. Elnagar, R. Al-Debsi and O. Einea, "Arabic Text Classification Using Deep Learning Models," *Information Processing and Management,* vol. 57, no. 1, pp. 103-114, 2020.

[15] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *ICML ,* 1997.

[16] A. Patra and D. Singh, "A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification Algorithms," *International Journal of Computer Applications,* vol. 75, pp. 14-18, 2013.

[17] M. Badieh Habib, T. Fouad Gharib and Z. Taha Fayed, "Arabic Text Classification Using Support Vector Machines. Arabic Text Classification Using Support Vector Machines," 2009.

[18] "Text Classification," [Online]. Available: https://monkeylearn.com/text-classification/.

[19] A. A. Elbery, "Classification of Arabic Documents," 2012.

[20] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed and A. Al-Rajeh, "Automatic Arabic Text Classification," 2008.

[21] A. Mesleh, "Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System," *Journal of Computer Science,* 2007.

[22] H. M. Noaman, S. Elmougy, A. Ghoneim and T. Hamza, "Naive Bayes classifier based arabic document categorization," INFOS2010 - 2010 7th International Conference on Informatics and Systems, 2010.

[23] A. Goweder, M. Elboashi, A. Elbekai and L. Academy, "Centroid-Based Arabic Classifier," 2013.

[24] K. VAIBHAV, "AUTOMATIC POETRY CLASSIFICATION USING NATURAL LANGUAGE PROCESSING," 2018.

[25] S. Kane-mainier, "How the Genre and Work of Poetry Are Represented," 2015.

[26] M. A. Ahmed and S. Trausan-Matu, "A Program For Analyzing Classical Arabic Poetry For Teaching Purposes," *Romanian Journal of Human - Computer Interaction,* vol. 10, no. 4, pp. 331-344, 2017.

[27] A. Almuhareb, W. A. Almutairi, H. Altuwaijri, A. Almubarak and M. Khan, "Recognition of Modern Arabic Poems," *Journal of Software,* vol. 10, no. 4, pp. 454-464, 2015.

[28] "Classical Arabic Poetry - Its form and theme - Arabic poetry in the early Renaissance".

[29] S. K. Jayyusi, "Umayyad poetry," in *Arabic Literature to the End of the Umayyad Period*, Cambridge University Press, 2012, pp. 387-432.

[30] A. Al-Falahi , M. Ramdani and M. Bellafkih, "Machine Learning for Authorship Attribution in Arabic Poetry," *International Journal of Future Computer and Communication,* vol. 6, no. 2, pp. 42-46, 2017.

[31] O. Alsharif and N. Ghneim, "Emotion Classification in Arabic Poetry using Machine Learning," *International Journal of Computer Applications,* vol. 65, no. 16, 2013.

[32] I. A. Mohammad, "NAIVE BAYES FOR CLASSICAL ARABIC POETRY," *Journal of Al-Nahrain University* , vol. 12, no. 4, pp. 217-225, 2009.

[33] R. M. Sallam, H. M. Mousa and M. Hussein, "Improving Arabic Text Categorization using Normalization and Stemming Techniques," *International Journal of Computer Applications,* vol. 135, no. 2, 2016.

[34] T. SRIVASTAVA, "Analytics Vidhya," 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. [Accessed 2020].

[35] Z. Lateef, "edureka," 2020. [Online]. Available: https://www.edureka.co/blog/decision-tree-algorithm/. [Accessed 2020].

[36] R. Saxena, "Dataaspirant," 2017. [Online]. Available: https://dataaspirant.com/support-vector-machine-algorithm/. [Accessed 2020].

[37] S. Polamuri, "Dataaspirant," 2017. [Online]. Available: https://dataaspirant.com/how-logistic-regression-model-works/. [Accessed 2020].

[38] M. Waseem, "edureka," 2019. [Online]. Available: https://www.edureka.co/blog/logistic-regression-in-python/. [Accessed 2009].

[39] S. Motwani, "Springboard Blog," 2020. [Online]. Available: https://in.springboard.com/blog/naive-bayes-classification/. [Accessed 2020].

[40] N. SHARMA, "Medium," 2020. [Online]. Available: https://medium.com/@nansha3120/bernoulli-naive-bayes-and-its-implementation-cca33ccb8d2e. [Accessed 2020].

[41] Z. Lateef, "edureka," 2019. [Online]. Available: https://www.edureka.co/blog/random-forest-classifier/. [Accessed 2020].

[42] H. Singh, "towards data science," 2018. [Online]. Available: https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab. [Accessed 2020].