SENTIMENT CLASSIFICATION ON ARABIC CORPORAS: PRELIMINARY RESULTS OF A CROSS-STUDY

Asmaa MOUNTASSIR (*), Houda BENBRAHIM (*), Ilham BERRADA (*)
asmaa.mountassir@gmail.com, benbrahim@ensias.ma, iberrada@ensias.ma

(*) ALBIRONI Research Team, ENSIAS
Mohammed 5 University, Souissi
Rabat, Morocco

Keywords:
Sentiment Analysis, Opinion Mining, Machine Learning, Text Classification, Natural Language Processing, Arabic Language, Corpora

Abstract:
The rise of social media (such as online web forums and social networking sites) has attracted interests to mining and analyzing opinions available on the web. The online opinion has become the object of studies in many research areas; especially that called “Opinion Mining and Sentiment Analysis”. Several interesting and advanced works were performed on few languages (in particular English). However, there were very few studies on Morphologically Rich Languages such as Arabic. This paper presents the study we have carried out to investigate supervised sentiment classification in an Arabic context. We use two Arabic Corpora which are different in many aspects. We use three common classifiers known by their effectiveness, namely Naïve Bayes, Support Vector Machines and k-Nearest Neighbor. We investigate some settings to identify those that allow achieving the best results. These settings are about stemming type, term frequency thresholding, term weighting and n-gram words. We show that Naïve Bayes and Support Vector Machines are competitively effective; however k-Nearest Neighbor’s effectiveness depends on the corpus. Through this study, we recommend to use light-stemming rather than stemming, to remove terms that occur once, to combine unigram and bigram words and to use presence-based weighting rather than frequency-based one. Our results show also that classification performance can be influenced by documents length, documents homogeneity and the nature of document authors. However, the size of data sets does not have an impact on classification results.

Résumé:
La montée des médias sociaux (tels que les forums web en ligne et les sites de réseaux sociaux) a attiré beaucoup d’intérêt à fouiller et analyser les opinions disponibles sur le web. Ainsi, l’opinion en ligne est devenue l’objet d’étude dans plusieurs domaines de recherche ; en l’occurrence le domaine dit « Opinion Mining and Sentiment Analysis ». Plusieurs travaux intéressants et avancés ont été menés sur peu de langues (notamment l’Anglais). Néanmoins, les langues dites Riches Morphologiquement, comme l’Arabe, ont connu très peu d’études dessus. Le présent papier rapporte les détails de l’étude que nous avons menée dans le but
d’investiguer la classification supervisée de sentiment dans un contexte arabe. Nous avons utilisé deux corpus arabes différents à plusieurs niveaux. Nous avons utilisé trois classificateurs standards et connus par leur efficacité, à savoir Naïve Bayes, Support Vector Machines et k-Nearest Neighbor. Nous investiguons un ensemble de settings pour en identifier ceux permettant de donner les meilleurs résultats. Les settings ainsi étudiés concernent le type de racination, le seuillage de fréquence des termes, la pondération des termes et les n-grammes mots. Nous montrons que Naïve Bayes et Support Vector Machines sont efficaces et compétitifs. Néanmoins, la performance de k-Nearest Neighbor dépend du corpus. Nous recommandons, à travers cette étude, d’utiliser la pseudo-racination plutôt que la racination, de supprimer les termes apparaissant une seule fois, de combiner les uni-grammes avec les bi-grammes mots et d’utiliser une pondération à base de présence plutôt qu’une pondération à base de fréquence. Les résultats de notre étude montrent également que la performance de classification peut être influencée par la longueur et l’homogénéité des documents ainsi que par la nature des auteurs des documents. Par contre, la taille des corpus n’a pas d’impact sur les résultats de classification.
1 Introduction

With the emergence of Web 2.0, internet users are more and more invited to express their opinions on the web through Social networking sites, online news sites, video-sharing sites, online web forums, personal blogs, online review sites, etc. These opinions have become so interesting for business bodies as well as for users. Companies are interested in knowing consumers’ feedback about their products. This could help them change their strategies to improve their products‘ quality as well as their services’ quality. They can also monitor their competitors’ reputation. Likewise, internet users can benefit from opinions available on the web when they need to buy a given product. By reading several comments about a product, this can influence their purchase decision. We can see that internet users are no longer simple web users; they are simultaneously consumers and producers regarding web content. This revolutionary change presents new challenges and, at the same time, great opportunities for researchers in different areas, in particular the research area related to Sentiment Analysis.

As a definition, Sentiment Analysis (called also Opinion Mining) is a research area that consists of mining in the web opinions expressed by internet users about a given subject. The comments that contain these opinions are collected in order to analyze them and perform a number of tasks on them. Among these tasks we find Subjectivity Analysis [1], Polarity Classification or Sentiment Classification [2], identification of attitudes and opinions intensity [3], and Summarization [4]. We stress that there are three levels of analysis; namely document, sentence and term-level. Working on document-level means that we are interested in identifying the polarity (positive or negative) of the whole document [2]. But if the analysis is conducted on sentence-level, the goal is to determine the polarity of each sentence in the document [1]. The work on term-level is known as Semantic Orientation [5]. It focuses on identifying the polarity of each term in the document. Note that we mean by document a text unit of analysis. In our study, documents are comments or posts written by internet users.

Most of research works performed in this field have been conducted on some European languages (especially English) and Asian ones (Japanese and Chinese). Nevertheless, very few works were performed on Morphologically-Rich Languages (such as Arabic, Hebrew and Czech) [6]. This is due to two major factors; the problem related to the lack of content available on the web on one hand, and the problem related to the complexity of processing these languages on the other hand. Indeed, and as a definition, a Morphologically-Rich Language (MRL) is a language in which significant information concerning syntactic units and relations is expressed at word-level [6]. The Arabic language presents an important case of MRL since it was judged by Internet World Stats as the language with the fastest growth rate, in terms of internet users, in the last 11 years1. We point out that Arabic language has three forms; namely Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA). CA includes classical historical liturgical text, MSA includes news media and formal speech, and DA includes predominantly spoken vernaculars and has no written standards [7]. In our study we deal with the MSA form.

The aim of this paper is to report details of the study that we have conducted in order to investigate Sentiment Analysis in the Arabic context. We have focused on classification by polarity on document-level, i.e. documents are classified as either positive or negative. For the classification task, we have applied three standard classifiers of the supervised approach; namely Naïve Bayes [8], Support Vector Machines [9] and k-Nearest Neighbor [10]. We have three main goals. First, a comparative study of the three classifiers in classifying sentiments in an Arabic context. Second, the investigation of a number of settings to identify the best of them. These settings concern stemming type, weighting scheme, term frequency thresholding and n-gram model. Third, the comparison between results obtained on two different Arabic corpora.

Among works that were performed on sentiment classification on document-level we find that of Dave et al. [11] who investigated sentiment classification on document level; their study was carried out on English product-reviews. Their classification task was based on some score words rather than technics of machine learning. They found that in some settings, bigrams and trigrams yield better product-review polarity classification. Pang et al. [2] have worked on English movie-reviews by using different features including n-gram words. They aimed to investigate the performance of three supervised classifiers (namely Naïve Bayes, Support Vector Machines and Maximum Entropy) in classifying sentiments. They found, on one hand, that unigrams outperform bigrams when classifying movie reviews by sentiment polarity. On the other hand, they concluded that the machine learning methods used do not perform as well on sentiment classification as on traditional topic-based categorization. Turney [5] has used an unsupervised approach based on the semantic orientation of document terms to determine the document polarity. His study was conducted on English product-reviews. The technic he used is called “Semantic Orientation using Pointwise Mutual Information”; it consists of computing for each word its score on the basis of likelihood of its co-occurring with the two words “poor” and “excellent” on the web. The semantic orientation of a sentence is obtained by averaging the scores of its words.

---

Concerning Arabic Sentiment Analysis, Abbasi et al. [12] have worked on document-level by using syntactic and stylistic features and a feature selection algorithm that they developed EWGA (Entropy Weighted Genetic Algorithm). Their work was conducted on comments of a Middle Eastern extremist group. Thanks to this algorithm, they achieved high results. Abdulmageed et al. [1] have worked on sentence-level to determine the document subjectivity (subjective or objective) by using Support Vector Machines as classifier and different types of features (including n-grams). Their study was conducted on the Arabic Tree Bank corpus. Almas and Ahmad [13] used an unsupervised approach based on local grammars to investigate Sentiment Analysis in financial newswires. Their study was domain-specific. The closest work to our study is that of Rushdi-Saleh et al. [14] who worked on the Opinion Corpus for Arabic (OCA), a corpus that they built from movie-reviews. They used Naïve Bayes and Support Vector Machines as classifiers and n-gram words as features to classify OCA documents on document-level. They found that stemming is not recommended in Arabic context. Moreover, they showed that it is possible to translate the Arabic corpus into English and perform classification on the translated version; the translation degrades slightly the classification performance.

The reminder of this paper is organized as follows. The second section deals with document preprocessing steps as well as document representation process. The third section describes the experimental environment and the performed experiments, then it discusses the obtained results. We finish by a conclusion and future works in the last section.

2. Document Preprocessing and Representation

Before we could classify a given document, it is essential to first apply the preprocessing process which consists of cleaning, normalizing and preparing the text document to the classification step. We describe below the most common methods of preprocessing phase.

- **Tokenization:** This process aims to transform a text document to a sequence of tokens separated by white spaces. The output of this process is a text free from punctuation marks and special characters. This is the basic definition of tokenization. However, this process takes a deeper dimension in Arabic language; it is called Segmentation [1]. The segmentation process consists of separating a word from its proclitics (including proclitics and enclitics) and the determiner Al. As a definition, proclitics are a kind of affixes that can be associated to Arabic words; they can be prepositions, conjunctions, future markers, etc. As examples, the segmentation of the word « الواعل » (work), and the segmentation of the word « يعتبرهم » (he considers) gives « يعتبر » and « يعتبرهم » (he considers). The first example illustrates proclitic removal while the second one is a case of enclitic removal.

- **Stemming:** For Arabic language, there are two different morphological analysis techniques; namely stemming and light-stemming. Stemming reduces words to their stems [15; 16]. Light-stemming, in contrast, removes common affixes from words without reducing them to their stems [17]. Stemming would reduce the words « درس » (the school), « مدرسة » (the teacher) and « الدراسة » (the study) to one stem « درس » (to study). While light-stemming would reduce the words « الدروس » (the studies) and « مدرسان » (two teachers) to respectively « دراسة » (a study) and « مدرس » (a teacher). The main idea for using light stemming [15; 16] is that many word variants do not have similar meanings or semantics although these word variants are generated from the same root. For example, the stemming of the two words « رائع » (wonderful) and « رهيب » (horrify) gives the word « رائع » (wonderful) and « رهيب » (horrify). We can see that the polarity of « رائع » is inversed by stemming. Hence, light-stemming allows retaining words’ meanings.

- **Stop Words removal:** Stop Words refer to function words (such as articles, prepositions, conjunctions, and pronouns) which provide structure in language rather than content, they refer also to words that do not have an impact on categories discrimination. Note that the list of stop words (called stoplist) is typically established manually, it is domain and language-specific.

- **Term Frequency Thresholding:** This process used to eliminate words whose frequencies are either above a pre-specified upper threshold or below a pre-specified lower threshold. This process helps to enhance classification performance since terms that rarely appear in a document collection will have little discriminative power and can be eliminated. Likewise, high frequency terms are assumed to be common and thus not to have discriminative power either.

Once the preprocessing stage accomplished, each document is mapped to a vector representation. We refer to the Vector Space Model (VSM). The set of terms (single words or expressions) retained after the preprocessing stage is called dictionary. These retained terms are called features. The document vector is obtained by computing the weight of each term with respect to this document. There are several weighting schemes; the most common are presence, frequency and TFIDF-based
weightings (more details about these concepts are given in section 4). The generation of such vectors means that the VSM of the corpus is established. Hence, the documents are ready to be as input for classification algorithms.

Let \( \{f_1, \ldots, f_n\} \) be the features set, and \( n_i(d) \) be the weight of feature \( f_i \) regarding document \( d \). Each document is represented by a vector \( d := (n_1(d), \ldots, n_m(d)) \). It is of worth to stress that, basically, the dimensionality of the native feature space is so strong; it can be tens or hundreds of thousands of terms for even a moderate-sized data set [18]. This is considered as a serious problem since it has three major drawbacks. First, it makes some classifiers intractable because of their complexity [19]. Second, it affects the effectiveness of classification since most of native features are noisy or does not carry information for classification [20]. Third, this makes classification so expensive. Therefore, it is essential to eliminate useless features and so reduce the dimensionality. There are several techniques for dimensionality reduction, including some common methods such as stemming, stop words removal and thresholding. Moreover, there are several feature selection algorithms that compute the “goodness” of each feature in order to evaluate how significant it is. Thanks to these algorithms, we could perform aggressive dimensionality reduction without losing in classification effectiveness. Among these algorithms we cite Mutual Information, Information Gain and Chi Square [18].

3. Experiments

In this section, we report details about the experiments that we have carried out. First, we present the data collections that we have used. Second, we describe the experiments’ environment, i.e. some details about the classifiers, the validation method and performance measures that we have used. The results of our experiments are reported and discussed at the end of this section.

3.1. Data collections:

In our experiments, we have used two Arabic corpora. The first corpus is ACOM that we developed internally. The second is OCA that we found freely available on the web. In the following, we present each of the two corpora.

3.1.1. ACOM:

As mentioned in the introduction, Arabic suffers from lack in digital content on the web. Indeed, we could not find Arabic sites that invite internet users to let or express their opinions such as IMDB (www.imdb.com) or Epinions (www.epinions.com). Moreover, and after mining in several Arabic web forums such as Maktoob (www.maktoob.com) and Koora (www.koora.com), we concluded that these forums could not be a good data source for our study for two reasons. On one hand, the Arabic used by internet users in these forums is generally dialectal; this could not help us as we are interested in MSA. On the other hand, we noted that for discussions, just the first post has an evaluable content, otherwise the replies are just greeting expressions like for instance « بارك الله فیك » (May God bless you brother).

The source that we judged suitable for our study was Aljazeera’s site\(^2\). In order to leave a comment, the user has to write in MSA according to the participation terms. We collected our data from Aljazeera’s polls and forums.

We called our corpus ACOM (Arabic Corpus for Opinion Mining). It consists of two data sets; each one is domain-specific. The first data set (called DS1) falls within movie-review domain, it consists of 594 documents. While the second data set (called DS2) is sport-specific data set, it consists of 1492 comments.

As the classification approach we are interested in is supervised, the whole of our corpus has to be manually annotated. There are four annotation categories, namely POSITIVE, NEGATIVE, NEUTRAL and DIALECTAL. We considered as positive each comment reflecting positive sentiments either toward the film topic, the actors or the story itself. It is the same for negative comments. In contrast, the neutral category consists of three types of comments, namely the comments where no opinion is expressed (as example a comment that reports just a quranic verse), the comments where the opinion expressed is neutral (as the expression « والله لا أدري» (I swear that I cannot distinguish what is right from what is wrong)) and comments that comprise a mixture of positive and negative opinions (as in « هذا المسلسل رائع لكن لا أريد تجسيد الصحابة » (This series is wonderful, but I cannot stand the incarnation of Prophet companions)). Finally, we classified as dialectal each comment that contains expressions in dialectal Arabic. This category used to clean our data sets from comments not written in MSA.

\(^2\) http://www.aljazeera.net
It is of worth to note that, during the annotation phase, we noticed that an opinion (positive or negative) is not often expressed explicitly, but can be expressed in a subtle manner. This is why the task of sentiment classification is more complicated than the other categorization types [21]. We give as examples these two comments that have been classified as negative « 'افدودوُذوَذوْيأف » (There is no power but from God) and « 'افدودوُذوَذوْيأف » (God is Sufficient for me; most Excellent is He in Whom I trust). We note also that, for DS2, we found that a great number of negative comments were ironic. If we do not take into account the discussion context while reading these comments, we would consider them as positive. But in fact they carry a negative sense. We can illustrate this case by this example « 'افدودوُذوَذوْيأف » (This is my humble opinion about your genius; I thank you dear professor because we benefited from your intelligence).

In table 1, we present the number of comments per category for each data set of ACOM corpus. As we can observe from this table, the dialectal comments of each data set represent less than 7% of the whole data set. This poor percentage proves that Aljazeera’s site is a good data source for our study. However, we can see that, for DS1, the negative comments are much larger than the positive ones. The reason of the negative comments predomination is perhaps the fact that Aljazeera’s topics are usually related to the Arab-Muslim world’s problems. This represents a drawback of Aljazeera’s comments.

### Table 1: Number of comments per category for each data set of ACOM

<table>
<thead>
<tr>
<th></th>
<th>POSITIVE</th>
<th>NEGATIVE</th>
<th>NEUTRAL</th>
<th>DIALECTAL</th>
<th>Total of Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>184</td>
<td>284</td>
<td>106</td>
<td>20</td>
<td>594</td>
</tr>
<tr>
<td>DS2</td>
<td>486</td>
<td>517</td>
<td>391</td>
<td>98</td>
<td>1492</td>
</tr>
</tbody>
</table>

As our study deals with a binary classification where the categories are POSITIVE vs. NEGATIVE, we have eliminated comments of NEUTRAL and DIALECTAL categories. The neutral documents will be introduced in next studies as they represent an important percentage (up to 26.2%) of the whole data set. Afterward, we proceeded to data set balancing with respect to categories distribution. We have eliminated a number of negative comments from each data set in a way to equalize the number of documents for each category. Table 2 gives more details about the resulted data sets; it shows for each data set the total number of documents, the total number of tokens and the average of tokens number per document.

### Table 2: Statistics on each data set of ACOM

<table>
<thead>
<tr>
<th></th>
<th>POSITIVE</th>
<th>NEGATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>Nb Doc</td>
<td>184</td>
</tr>
<tr>
<td></td>
<td>Nb Tokens</td>
<td>11088</td>
</tr>
<tr>
<td></td>
<td>AVG Tokens</td>
<td>60,26</td>
</tr>
<tr>
<td>DS2</td>
<td>486</td>
<td>27733</td>
</tr>
<tr>
<td></td>
<td>57,06</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NEGATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>Nb Doc</td>
</tr>
<tr>
<td></td>
<td>Nb Tokens</td>
</tr>
<tr>
<td></td>
<td>AVG Tokens</td>
</tr>
<tr>
<td>DS2</td>
<td>514</td>
</tr>
<tr>
<td></td>
<td>66,41</td>
</tr>
<tr>
<td></td>
<td>Total Documents</td>
</tr>
<tr>
<td></td>
<td>Total Tokens</td>
</tr>
</tbody>
</table>

### 3.1.2. OCA:

OCA (Opinion Corpus for Arabic) is a corpus developed by Rushdi-Saleh et al. [14]. It consists of 500 movie-reviews collected from several Arabic blog sites and web pages. These documents were annotated automatically on the basis of the ratings associated to the reviews. Table 3 gives some statistics about the OCA corpus.
As we can see from tables 2 and 3, DS1, DS2 and OCA have different sizes. Moreover, we conclude that OCA comments are much longer than ACOM’s ones. So, in this study, we will have the opportunity to deal with different data sets with respect to document number and document length.

3.2. Classification and Algorithms:

In this study, we are interested in a binary classification, where each document is assigned to one of the two categories POSITIVE vs. NEGATIVE.

In order to perform our tasks of preprocessing and classification, we have used the data mining package Weka\(^3\) \cite{weka}. We have used three standard classifiers; namely Naïve Bayes (NB), Support Vector Machines (NB) and k-Nearest Neighbor (k-NN). The philosophy behind these three algorithms is quite different. However, each of them was shown as effective for several classification tasks \cite{naive,bayes,support,nearest}. For the NB classifier, we have used a kernel estimator, rather than a normal distribution, for numeric attributes \cite{kernel}. Concerning SVM classifier, we have used a normalized polynomial kernel with a Sequential Minimum Optimization (SMO) \cite{smo}. Furthermore, the k-NN classifier that we have used is based on a linear search with a cosine-based distance \cite{cosine}. Among the values of k tested for each setting, we choose those that allow achieving the best results. The selected values are various within the settings, they arrange from 1 to 23 in the present study.

We specify that the pre-mentioned selected parameters, about algorithms, were defined after several tests to determine the parameters that allow achieving the best results.

3.3. Evaluation:

3.3.1. Validation method:

The validation method that we have used is k-fold cross-validation \cite{crossvalidation}. It consists of partitioning the original sample into K sub samples. Of the K sub samples, a single sub sample is retained as the validation data to test the model, and the remaining K−1 sub samples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K sub samples used exactly once as the validation data. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. In our experiments, we have used 10-fold cross-validation.

3.3.2. Evaluation measure:

The category assignments of a binary classifier can be evaluated using a two-way contingency table (Table 4) for each category, which has four cells:

- cell a: counts the documents correctly assigned to this category;
- cell b: counts the documents incorrectly assigned to this category;
- cell c: counts the documents incorrectly rejected from this category;
- cell d: counts the documents correctly rejected from this category.

Table 3: Statistics on OCA

<table>
<thead>
<tr>
<th></th>
<th>POSITIVE</th>
<th></th>
<th>NEGATIVE</th>
<th></th>
<th></th>
<th>Total</th>
<th>Total Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nb Doc</td>
<td>Nb Tokens</td>
<td>AVG Tokens</td>
<td>Nb Doc</td>
<td>Nb Tokens</td>
<td>AVG Tokens</td>
<td>Documents</td>
</tr>
<tr>
<td>OCA</td>
<td>250</td>
<td>121392</td>
<td>485.56</td>
<td>250</td>
<td>94556</td>
<td>378.22</td>
<td>500</td>
</tr>
</tbody>
</table>

\(^3\) [http://www.cs.waikato.ac.nz/ml/weka/](http://www.cs.waikato.ac.nz/ml/weka/)
Table 4: Contingency Table

<table>
<thead>
<tr>
<th>Assigned number</th>
<th>C1 is correct</th>
<th>C2 is correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigned C1</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Assigned C2</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

Conventional performance measures [29] are defined and computed from these contingency tables. Among these measures we find precision (p) and recall (r). We can define precision and recall for class C1 as follows:

\[ p = \frac{a}{a + b} \text{ and } r = \frac{a}{a + c} \]

As we can see, precision is the ratio between the number of documents correctly classified and the number of all documents affected to the category. Recall is the ratio between the number of documents correctly classified and the number of documents that belong to the category. For a given classifier, we can consider the precision as its “strength degree”, while the recall can be as its “exhaustiveness degree”. These two measures are complementary.

We can combine precision and recall by using F1 measure; a metric defined by van Rijsbergen [30]. This is a common choice for a single-numbered performance measure; it is defined as the harmonic-mean of precision and recall:

\[ F_1 = \frac{2rp}{r + p} \]

F1-measure balances recall and precision in a way that it gives them equal weight. This measure is often used as an optimization criterion in threshold tuning for binary decisions. Its score is maximized when the values of recall and precision are equal or close; otherwise, the smaller of recall and precision dominates the value of F1.

For evaluating performance average across categories, there are two conventional methods, namely macro-averaging and micro-averaging [25]. Macro-averaged performance scores are computed by first computing the scores for the per-category contingency tables and then averaging these per-category scores to compute the global means. Micro-averaged performance scores are computed by first creating a global contingency table whose cell values are the sums of the corresponding cells in the per-category contingency tables, and then use this global contingency table to compute the micro-averaged performance scores. There is an important distinction between macro-averaging and micro-averaging. Micro-average performance scores give equal weight to every document, and are therefore considered a per-document average. Likewise, macro-average performance scores give equal weight to every category, regardless of its frequency, and is therefore a per-category average.

For our experiments, we have used as performance measure the Macro-average of F1 measure.

3.4. Experimental Design:

The present paper has three major goals. The first is about identifying the best settings for sentiment classification on Arabic corpora. These settings are about stemming type, weighting scheme, term frequency thresholding and feature type (n-gram words). The second goal is related to a comparison, in an Arabic context, between three common classifiers known by their effectiveness in text categorization. The third is about comparing the classification effectiveness obtained on three different sized Arabic data sets. We note that, for all tested settings, stop words were removed.

As we advanced in section 2, there are two types of stemming in Arabic, namely stemming and light-stemming. In this paper, we investigate the contribution of each type in enhancing the classification results.

The weighting feature types that we studied are three:
• **Presence:** This weighting (called also binary weighting) focuses on presence rather than frequency term regarding a given document. It consists of assigning 0 as weight to a term if it does not occur in the document, otherwise 1 is assigned. This weighting was introduced originally in sentiment analysis by Pang et al. in [2].

• **Normalized Frequency:** This weighting derives from frequency weighting, where each term is given as weight its frequency (number of occurrences) regarding a given document. We normalize these weightings by dividing them by the length (number of terms) of the document. We use normalization to avoid problems that can result from differences in documents size. Indeed, a term can occur many times in a long document, but few times in a short one. The importance of this term can be the same regarding the two documents, while, if we consider a frequency weighting, this term would not have the same importance regarding the two documents.

• **TFIDF:** This term refers to Term Frequency/Inverse Document Frequency [29]. For a term t, the TFIDF weight is assigned the value:

\[
\text{TFIDF}(t) = \text{TF}(t) \times \text{IDF}(t),
\]

where TF represents the frequency of the term t (number of occurrences in the data set), and IDF measures the significance of this term. It is obtained by the following formula:

\[
\text{IDF}(t) = \log\left(\frac{n}{n(t)}\right),
\]

where n represents the total number of documents in the data set, and n(t) denotes the number of documents where term t occurs.

The higher is the value of TFIDF for a term, more it is considered significant.

For term frequency thresholding, we were interested in the minimum frequency. The values we considered were 1, 2 and 3. The idea is to eliminate from dictionary the term whose frequency (number of occurrences in the data set) is strictly less than the specified threshold. The objective was to find the best threshold for the three classifiers.

Concerning feature types, we focused on n-gram words [31]. As a definition, an n-gram word is a sequence of n terms in a given document. If n is equal to 1, we talk about unigrams, if it is equal to 2, they are called bigrams, and so on. For example, in the sentence « هذا المسلسل رائع » (this series is wonderful), the unigrams that we can extract are « هذا المسلسل رائع » and « رائع ». While the bigrams that we can find are « هذا المسلسل » and « المسلسل رائع ». In this study, we first consider the use of unigrams alone. Afterward, we investigate the adding of bigrams. Finally, we examine the combination of unigrams, bigrams and trigrams. We specify that n-gram extraction is performed before stemming and stop words removal.

In order to investigate these four settings (parameters), we change each time the value of a given parameter and we keep fixed values for the other parameters.

### 3.5. Results:

#### 3.5.1. Preprocessing Results:

The application of each setting generates a different dictionary, and so a different dimensionality, for each data set. Figure 1 gives an overview on the number of features obtained following the application of each setting. We can see through this figure how much stemming and term frequency thresholding contribute to the reduction of feature dimensionality on one hand, and how much n-gram adding enriches the feature space. We can also see the difference between ACOM’s dimensionality and the one of OCA.
3.5.2. Classification results:

In figures 2, 3 and 4, we present the obtained results (in terms of macro-averaged F1) of the three classifiers for the different settings for respectively DS1, DS2 and OCA data sets. We specify that, in these figures, “Presence”, “NormFreq” and “TFIDF” denote the three weighting schemes that we have tested. “MinFreqN” means that we apply a thresholding of minimum frequency; N is the value of the threshold. “uni” and “bi” correspond respectively to unigrams and bigrams.
Figure 2: Results in F1 measure for DS1

Figure 3: Results in F1 measure for DS2
The first test is about stemming. According to figures 2, 3 and 4, we can see that, generally, light-stemming gives better results than stemming. This result is logic since light stemming allows retaining words meanings. Hence, it is predictable that light-stemming performs well than stemming. This is why we keep applying light-stemming for the following tests. We note that, to our best knowledge, there was no comparative study performed between stemming and light-stemming in Sentiment Analysis tasks.

The second test deals with thresholding; we remove terms that occur once in the data set (MinFreq2), then we remove those that occur twice (MinFreq3). We can observe, through figures 2, 3 and 4, that the elimination of terms that occur once (these terms are called hapaxes) improves clearly the results in comparison with the non-application of thresholding. However, the removal of terms that occur twice does not enhance the results. This can be explained by the fact that removing hapaxes means that we eliminate, among others, mistakes made by the authors who are just simple internet users. Indeed, these mistakes can be a source of noise for classification. Nevertheless, the elimination of terms that occur twice may remove significant features. This is why we keep applying the thresholding with a threshold equal to 2 for the rest of tests.

The following test concerns the weighting schemes. As we can see from figures 2, 3 and 4, presence weighting performs well for all classifiers. We can also see that TFIDF weighting allows to SVM and k-NN achieving good results. However, we note that weighting based on normalized frequency gives poor results for all classifiers. Hence, we keep using the presence weighting for next tests. It is of worth to note that this result is consistent with that reported by Pang et al. [2]. Pang and Lee gave an interpretation to this finding: “While a topic is more likely to be emphasized by frequent occurrences of certain keywords, overall sentiment may not usually be highlighted through repeated use of the same terms” [21].

The last test has as goal to investigate the use of n-gram words. Figures 2, 3 and 4 show that adding bigrams to unigrams improves considerably the results in comparison with the use of unigrams only. However, and when we add trigrams, the performance is not enhanced for DS1 and DS2, but for OCA it is slightly improved. As the combination of unigrams, bigrams and trigrams is expensive and as it does not give considerable improvement, we keep the combination of unigrams and bigrams as feature model. We can interpret the contribution of bigrams to enhancing the results by the fact that, in contrast to unigrams, bigrams allow the
management of negation. Indeed, by considering this example « لا أافق » (I do not agree) which has a negative sense, if we consider unigrams only we will have just « أافق » (I agree) since the word negation « لا » is a stop word. We can see that, as a consequence, the polarity is inversed. But if we add bigrams, the expression « أافق لا » (I do not agree) will be kept as it is, and so the polarity is maintained.

We think that, in the context of Sentiment Classification, the negation treatment is the only contribution of bigram adding in comparison with the use of unigrams only. Indeed, to identify and classify sentiments, we do not need to identify some specific key words such as “Text Mining” and “Computer Science” that could be helpful in Text Categorization. This finding, about the combination of unigrams and bigrams, is different from the result reported by Pang et al. [2] since they found that the use of unigrams gives the best results. Our finding is also different from that reported by Rushdi-Saleh et al. [14] who stated that the obtained results with bigrams and trigrams were very similar to unigrams.

So far, the best setting identified for all classifiers and all data sets is the application of light-stemming, the elimination of hapaxes, the combination of unigrams and bigrams with the use of a presence-based weighting.

3.5.3. Discussion:

As a comparison between the three classifiers, we can observe through figures 2, 3 and 4 that the results regarding the classifiers performance are not the same. Indeed, for OCA, the three classifiers seem to be competitive in all performed tests. The best results achieved are 91% by SVM, 92.2 by NB and 93% by k-NN. However, for ACOM (DS1 and DS2), just NB and SVM are competitively effective; k-NN seems to be less effective since it yielded poor results for almost all tests. The best results achieved, for ACOM, are 87.5% by NB, 78.2% by SVM, and 74.4% by k-NN. Hence, we can conclude that NB and SVM are the most effective. But we cannot draw a conclusion about k-NN performance since its behavior was not the same regarding the two corpora. Yet, we can give an interpretation to this observation. K-NN yielded high results on OCA perhaps because this data set is homogeneous. In other words, the documents of this data set belong purely to the movie-review domain; so the documents of each category can be very similar. As the k-NN classification is based on similarity, we can understand why this classifier performed well on this corpus. However, we can say that k-NN did not give good results on DS1 because the documents of this data set are not homogeneous. In fact, these documents were collected from a discussion about a historical film that has generated much noise in the Arab-Muslim world. So the documents domain is not purely movie-review but also religion and history. It is the same for DS2 since its documents were collected from discussions about 18 topics of sport.

To compare between results obtained on the three data sets (DS1, DS2 and OCA), we can say that the classifiers yielded better results on OCA in comparison with ACOM. Indeed, the best results were achieved on OCA (up to 93% with k-NN). DS1 was in the second rank with up to 87.5% (a result achieved with NB). Finally, DS2 had the worst results with up to 76.4% (a result obtained by NB). We can explain this difference in results by authors’ nature for each corpus. In fact, documents OCA are well written in comparison with those of ACOM which contain a large number of mistakes. This is due to the nature of document authors; documents of OCA are produced either by bloggers or professional writers, while documents of ACOM are, generally, posted by mere internet users. Furthermore, documents of OCA are much longer than ACOM’s ones (see tables 2 and 3). So we can draw the conclusion that more the documents are long, more the classification is effective. This finding remains to be verified in future studies. There is another different aspect, between the two corpora, that concerns annotation. We recall that documents of ACOM were manually annotated by one annotator, while documents of OCA were automatically annotated on the basis of rating systems (see [14] for more details). We are not sure that annotation method could have an impact on classification performance. This detail is also to be further studied in future works. Finally, we recall that a great part of negative documents of DS2 were written in an ironic manner, i.e. they seem to be positive while, in fact, they carry a negative sense. This type of documents represents a major challenge for classification. This may explain why this data set (DS2) had the worst results.

We point out a surprising result about the effectiveness obtained on OCA. Recall that this corpus is a collection of long movie-reviews. These reviews does not contain only opinions, there are large parts that report some events of movies in question. These parts are considered as facts; they represent the objective parts of the documents. In sentiment classification, the objective parts do not give any contribution to classification performance. On the contrary, they have to be eliminated to clean documents since we are interested just in subjective parts; this is what we call Subjectivity Analysis. What was surprising is the fact that the results obtained on OCA were so high even there was no elimination of objective parts.

Concerning the impact of data set size on classification, we can say that the effectiveness of classification does not depend on the number of documents in the data sets. The size of DS2 (1000 documents) is the double of that of OCA (500 documents), and the size of OCA is larger than that of DS1 (368 documents), however the best results were obtained on OCA, then the ones on DS1; DS2 was the worst in terms of classification results.
We can say that our results (up to 93% in F1 measure) is competitive with those reported in the literature. Abbasi et al. [12] report a result of 93.6% in accuracy. Recall that they have used a feature selection algorithm that helps to eliminate non-informative features and hence improves considerably the results. Abdul-mageed et al. [1] achieved a result of 95.52% in F1 measure, but their work cannot be directly compared to ours since their work was on sentence-level, while ours was on document-level. Moreover, their corpus consists of newswire articles, while ours consists of comments written by simple internet users. Rushdi-Saleh et al. [14] report a result of 90.73% in F1 measure, recall that their experiments were carried out on OCA.

4. Conclusion and future works

The study that we have reported in this paper had as objective to investigate sentiment classification in an Arabic context. We carried out our study on two Arabic corpora with different sizes, namely ACOM and OCA. ACOM is a corpus that we have developed (from Aljazeera’s site) and annotated manually with two main categories: POSITIVE and NEGATIVE. It consists of two data sets; DS1 which is a collection of 368 movie-reviews and DS2 which is a collection of 1000 comments from sport domain. OCA is a collection of 500 movie-reviews collected by Rushdi-Saleh et al. [14]. We aimed to, first, investigate some settings to find those that yield the best results. The considered settings were about stemming type, term frequency thresholding, term weighting, and n-gram words. We used for classification three common classifiers known by their effectiveness, namely Naïve Bayes, Support Vector Machines and k-Nearest Neighbor. The second goal of this study was to compare the behavior of these three classifiers in an Arabic context. The third goal that we chase was about the comparison between the results yielded on the two corpora. The obtained results show that the best setting for almost all classifiers on all data sets was the application of light-stemming, the elimination of hapaxes (as thresholding), the combination of unigrams and bigrams, and the use of a presence-based weighting. We note that the TFIDF-based weighting is also a suitable weighting for both SVM and k-NN. To compare between the three classifiers, results show that they did not have the same behavior toward the two corpora. Indeed, for ACOM, NB and SVM were competitively effective (up to 87.5%), but k-NN was less effective (the best result recorded was 74.4%). Nevertheless, for OCA, the three classifiers showed a high effectiveness and were competitive along the performed tests (up to 92.2% for NB, 91% for SVM and 93% for k-NN). As a comparison between the two studied corpora, we found that the best results were obtained on OCA. This is due to the differences between the documents of the two corpora, namely the nature of document authors, document length and document topics. We show also that the size of data sets does not influence the performance of classification.

As future works, we look forward to further studying and verifying the different findings reported in this paper. These findings are about the factors that can influence classification performance, namely documents length, documents homogeneity, the nature of authors and annotation method. We will also verify the assumption about the non-influence of data set’s size on classification effectiveness. We look forward to apply subjectivity analysis on OCA to study the effect of this analysis to enhance classification. Moreover, we would like to introduce other types of features (rather than n-gram words), such as n-gram letters, Parts Of Speech and features that depend to Arabic language. We will also use features for handling negation (such as the association of the word NOT to the negated tokens), this might help us to avoid the use of bigrams since they are expensive. Finally, we plan to apply random sampling, rather than 10-fold cross-validation, on the non-balanced DS1.

5. References


