ISAO: An Intelligent System of Opinions Analysis

Sarra Zrigui[1], Rami Ayadi[2], Anis Zouaghi[3], Salah Zrigui[4]

[1,3] ISSAT Sousse, Tunisia, LATICE Laboratory
sara.zrigui@gmail.com, Anis.zouaghi@gmail.com,
[2]LaTICE laboratory, University of Sfax, Sfax, Tunisia
ayadi.rami@planet.tn
[4]École Nationale Supérieure d'Informatique et de Mathématiques
Grenoble, France
salahzrigui@gmail.com

Abstract. Today, the need to automatically process opinions is strongly felt. It is in this context that we situate this work whose objective is to contribute to the achievement of opinions analysis system, enabling a binary classification on a set of textual data. For this, we studied and evaluated several methods, Support Vector Machines (SVM) and Naïve Bayes (NB), on a corpus composed of 500 journals films. These models have not been satisfactory. To improve the results we have introduced a pre-treatment phase or standardization corpus before classification; this phase has improved the quality of the classification.

Keywords: opinions analysis, Arabic, classification, SVM, NB.

1 Introduction

Nowadays, the amount of information generated by users is increasing very rapidly. Whether on forums, blogs, e-commerce sites or social media sites, users continue to share their knowledge and their views on products, ideas, events, etc. This large amount of opinions can influence the way to perceive brands, people, organizations and events, which can motivate the masses to action. Hence the need to create systems for classification and analysis of opinions was born. To meet this need, many researches have emerged. They come from different areas: data mining, decision support, knowledge modeling, natural language processing etc.

The social internet such as social networks, forums, blogs, e-shopping sites, etc. detonated the number of texts expressing opinions. Millions of messages appear every day in social networks such as Twitter and Facebook for example.

Increasingly such sites are used by users to post their opinions about products and services they use, or express their political and religious views. These websites have become a very valuable source of opinions and feelings of people. This social data can then be used for:

- Marketing by analyzing the views of users about a product or brand.
- Social studies analyzing the societal trends. etc.

The need to automatically process opinions is therefore strongly felt. It is in this context that this work is inscribed, and whose objective is:

- To contribute to the realization of an opinions analysis system, enabling a binary classification on a set of textual data.
- To study and evaluate the effectiveness of support vector machine (Support Vector Machines (SVM) in English) and the naive Bayesian model (Naïve Bayes (NB)) to perform this task.
- To consider and test the impact of a number of pretreatments on the analysis of views and system performance.

This paper consists of five sections; in the second section we'll describe the Arabic language and its complexity. In the third, we present some related work. In the fourth section we present our contribution. Finally we end this paper with a conclusion.

2 Problems of Opinion Analysis in Arabic

2.1 Arabic Language

Arabic is the fifth most used language in the world. It is the mother tongue of over 200 million people and more than 450 million speakers [1]. The Arabic language is considered by Internet World Stats [2] as the language with the fastest growth rate in terms of internet users in the last eleven years. The Arabic language has three forms; namely Classical Arabic (CA), Modern Standard Arabic (MSA), and the Arabic dialect (DA). CA is one used in conventional historical texts, the MSA is the language used by the media and in official speeches, and finally the DA consists mainly dialects spoken and has no written standards [3]. The Arabic alphabet consists of 28 letters, unlike Latin alphabets, the orientation of the Arabic writing is from right to left. Unlike the English language, for example, the notion of upper and lower case does not exist in Arabic.

2.2 Complexity of Automatic Processing of the Arabic Language

Arabic is a difficult language to automatically deal with for several reasons, among which we can mention [4]:

- The presence of diacritics makes it a less ambiguous and more phonetic language, but unfortunately the majority of texts are not vowelized.
- Certain combinations of characters can be written in different ways.
- A very complex morphology compared to the English language.
- Synonyms are widespread. The Arabic language is a highly inflected language and derivational.
- Lack of publicly available Arabic corpora.
- Lack of Arabic digital content.

Meaning of words. A word can have more than one meaning depending on the context in which it is used. A word can have more than one lexical category (noun, verb, adjective, etc.) in different contexts. The following figure shows an example of this ambiguity.

Synonyms. There are many words that are considered synonymous. Given a corpus, researchers can use the tools of morphological analysis to find synonyms of a word, the frequency of each word of these synonyms and if one of them is more common

Form of words depending on its mode of use. The form of some Arabic words can change according to their case modes [16] (nominative, accusative or genitive). For example, the plural of a word (مسافر) meaning (traveler) may be in the form (مسافرون) in the case of nominative (مرفوعة) and shape (مسافرين) in the case of the accusative / genitive (منصوبة / مجرورة).

Morphological characteristics. An Arabic word consists of a root, more affixes and clitics. The stem consists of a consonant root (صحيح جذر) and a pattern of morpheme (معني ذات كلمة اصغر). Affixes include time markers, sex and / or numbers (اعرابية حركات). Clitics include some prepositions (الجر حروف), conjunctions (العطف حروف) determinants (داتمحد), possessive pronouns (الملكية ضمائر) and pronouns (ضمائر) [17]. The Stemming process reduces the number of features extracted from a corpus by converting the words to their stems. There is another approach to the reduction of the morphology which simply removes affixes and does not convert the word to its stem. This approach is called Light Stemming.

3 Related Work on Arabic

The work done in the context of Arabic is limited to the work performed on the application of different classification techniques and also of the work on the pre-treatment applied on the text before the classification process.

In this paper [5], Al-Kabir, address the issue of the effect of the Stemming the classification of Arabic text documents. It applies the text classification for text documents using Stemming in the pretreatment steps. The results showed that the support vector machine (SVM) classifier has reached the precision of the highest classification using the two test modes with 87.79% and 88.54%.

The main objective of this study [6] is to measure the accuracy for each classifier to determine which is more accurate for the Arabic text classification based on function words. Classifiers are studied Support Vector Machine (SVM) with sequential optimization Minimal (SMO), Naive Bayes (NB), and J48. The results show that the use of SMO provides the highest accuracy and lowest error rate, and that the time needed to build the SMO model is the smallest time.

There are several studies that compare the performance of different classification algorithms on Arabic text. In [7], Alsaleem studied the performance of methods, Bayesian Naïve (NB) algorithm and Support Vector Machine (SVM), on different sets of Arabic data. The experimental results against various Arab text data sets show that SVM algorithm outperforms the NB with regard to all measures.

In [8], the author compared the performance of KNN and SVM. This study showed that both have superior performance, and SVM improved accuracy and time. In [9], the author have applied KNN and NB on the text in Arabic and have concluded that KNN has better performance than NB, they also concluded that the selection of features and the size of the overall training and value K affected the performance of the classification.

EL-HALEES [10] compared six well-known classifiers applied to the Arabic text; ANN, SVM, NB, KNN, maximum entropy and the decision tree. It showed that NB and SVM classifiers are the best in terms of F-measure with 91% and 88% respectively. In [11], Al-Khorsheed thubaity studied a variety of text classification techniques; SVM, Knn, NB using the same data sets that belong to a wide range of categories.

4 Experiment and Evaluation

In this part we will present the experimental study. We will start by showing the tools that have contributed to this work and the approach.

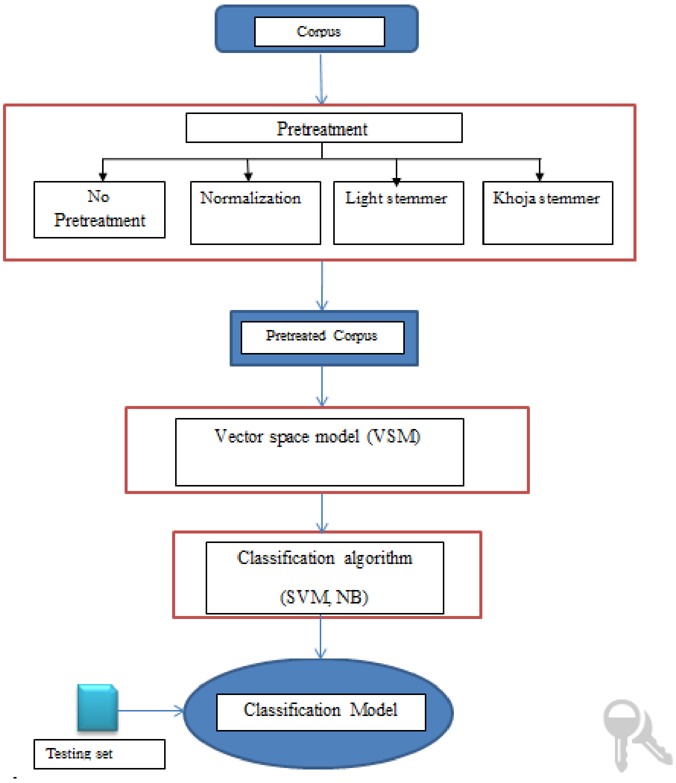

Fig. 1 Classification process

4.1 Preprocessing

Generally natural language texts cannot be directly analyzed (interpreted for example by a classifier or by classification algorithms)[18]. Whatever type of data, it is necessary to pretreat the raw data in order to then treat it with unified processes and not with a multitude of processes adapted to all possible cases [12].

These pretreatments are done to standardize the different ways of writing the same word to get correct results, to correct obvious spelling or typographical inconsistencies and to clarify certain lexical information implicitly expressed in the texts and some missing information to the using external resources.

It is necessary to use these treatments according to the final objective in order clarification or maximize the number of operations performed. Many of these treatments are specific to the used language (it is not the same type of pretreatment to documents written in English and French or Arabic).

At the simplest level, pretreatment is to index and count all the words found in the entry documents to calculate a table of documents and words, ie, a frequency matrix that lists the number of appearances of each word. This basic process can be refined to exclude some common words such as "الذي" and "ذلك" (empty word) and combine different grammatical forms of the same words such as "مسافرين" ,"مسافرون" ,"مسافر" etc.

In this step, the Arabic texts are transferred to a format suitable for Stemming process. Generally, punctuation and special characters are removed. This is followed by the application of certain linguistic processing. Some of the most popular treatments are [13][15]:

Each document in the set of data is processed to Arabic
Delete all numbers and punctuation marks.
Remove all vowels except (. (ة د ش ل ا
Duplicate all the letters containing the symbols (shadda).
Converting "أ" ,"إ" and "آ" to "ا".
Convert "ى" to "ي" and "ة" to "." ه
All non-Arabic words are filtered.
Arabic empty words are removed.
Apply a Stemming Algorithm.

In the sequel, we present the algorithms we used for the pretreatment of the data.

Removing unnecessary characters. Removing unnecessary characters is performed by following the steps of the following algorithm:

```
Algorithm 1 : Removing useless characters
d a document of N word w d=(w1,…,wn)
Input: text
Output: text without useless characters
For each word wi of d do
        1. Delete all punctuation marks
        2. Delete all latin numbers and characters
        3. Delete all abbreviations and isolated letters
```

- Punctuation: This step will remove any sequence of punctuation characters delimited by letters or spaces such as the comma and semicolon ... etc. In Arabic texts, some characters are written from right to left as the question mark "?"And the comma", ", this orientation is also taken into account during processing.
- The numbers and Latin characters: Here all character sequences located between two spaces containing Arabic numerals "1 ... .9" or Latin numbers or Roman «1 ... .9 ... I. IX" are eliminated, and we also eliminate the Latin letters "A ... Z, a ... z".
- The abbreviations and single letters: The abbreviations of words, such as: ت for "تاريخ = date" م to ميلادي ,ص for "صفحة = page" ج for "جواب = answer," س for "سؤال = question" . or coordination as ب, و, ف, ل ك (bi-, wa, fa-, li-, ka ...) scored as isolated forms next to the numbers (e.g. 5 ب 32) or the mathematical formulas as (3 + ع س =).

Removal of empty words. The empty words are the words that frequently occur in most documents in a given collection without significant semantic relation to the context in which they exist. They will be removed from the text because their presence or absence does not provide useful information on the meaning of the text. The figure 2 contains a partial list of empty words. In Arabic, the list of stop words can include the punctuation marks (!? ...), Pronouns (... هم التي الذي هي هو), adverbs (... فوق تحت أمام خلف), days of the week (... الاربعاء الثلاثاء الاثنين) months of the year (... نوفمبر اكتوبر سبتمبر). There is no definitive list of empty words used in all tools. They are also tools that use no stoplist. Some tools specifically prevents their use to support the search phrase.

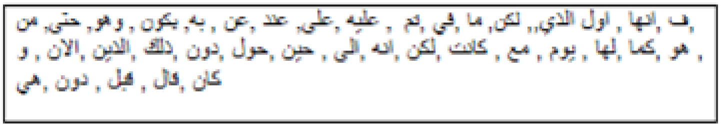
ف, انها , اول الذي,, لكن, ما, في, ثم , عليه, على, عند, عن , به, يكون , و هو, حتى, من
, هو, كما, لها , يوم , مع , كانت, لكن, انه, الى , حين, حول, دون, ذلك, الذين, الان , و
كان, قال , قبل , دون, هي

Fig. 2 partial list of empty words

Morphological processing. The morphological character normalization was achieved by following a number of rules. These rules are defined by:

```
Algorithm 2 : Morphological processing
Input: text
Output: text after morphological traitments
For each word $w_i$ of $d$ do
        1. Convert all « إ » ,« أ » and « آ » to «ا »
        2. Convert all “ى” to “ي” and “ة” to “ه”
        3. Delete the character ‘—’
        4. Delete all vocalization signs : «ً,ٍ,ْ,ِ,ُ,َ»
        5. Duplicate characters with « ّ »
```

- The first step of this algorithm is to normalize "Alif and Hamza", it is to convert el "أ" ,"إ" and "آ" to "ا". The reason for this conversion is that all forms of Hamza are represented in dictionaries "ا", as most of the texts neglect adding Hamza El Alif and often ill people spell the different forms of aleph.

- The second step is standardizing ي and ة "Yâ' and el tâ marbouta". The 'Hamza' character adds to the confusion, whether it is at the end of the word, between ي (Letter yâ final) and ى (or 'alif maqsura): The word نادي, Nadi, "club", can be noted نادى, (read as nâdâ, "invite ").
- The third step is to delete the character '—' (kashida) because typesetters make frequent use of the '—' character (called kashida). These characters lengthen the line in the middle of words which allows for a clearer readability and reduce white space on a line or justified for purely calligraphic reasons. This character, not part of the Arabic alphabet, is often a source of confusion for the treatment of texts.
- In the last two steps we remove the signs of vocalization, all signs of vocalization ٌ, ِ, ُ, ِ, َ,are eliminated except for "" where we duplicate the character that contains it.

4.2 Results

Data set. The corpus used is the OCA corpus (Opinion Corpus for Arabic), developed by Rushdi Saleh and al [4].This corpus consists of 500 reviews for movies collected from various Arabic website and blog, classified respectively 250 250 positive reviews and negative reviews.

Tools. Several resources available on the web are ready to be used. We used a different combination of several tools developed in other research.

For pretreatment: Regarding the stemmers used the light stemmer Arabic and Arabic stemmer, the code is written in Java and published on the Internet developed by Mr. Saad [4].

For representation: to model the corpus, we also used the famous weka [14].

For classification: We used weka that implements a large collection of machine learning algorithms for data mining tasks.

Experiments. For the validation of our work, first preprocessing is performed to normalize each document in the corpus.

We made several experiments were prepared four combinations ready data polarity detection phase:

- The corpus without pretreatment.
- The corpus + Application of Standard
- The corpus + Application of arabic light stemmer
- The corpus + Application of Khoja stemmer
- Then we made a representation in words vector with the Weka tool.
- Then the classification, also using weka, is performed using these two classifiers SVM, NB.

The performances are evaluated using three metrics (precision, recall and F-measure).

In our case, 80% of the data is used for the training set and 20% is for the test. The following sections show the results obtained using this method.

We found that the best result was obtained using the method NB with a percentage of 82% classified document correctly, and that by the second experiment, the corpus + application standardization.

We still noticed that the application of the arabic light stemmer and Khoja stemmer degraded performance even compared to the result obtained using the corpus in its raw state. The following figure shows this clearly.

For the SVM classifier, we note that the best result was obtained with the corpus in the raw materials with 87% classified correctly. We also note that the application of arabic light stemmer and Khoja stemmer degraded performance. The results are displayed in the following figure.

The comparison between the results obtained by NB and SVM, shown in the table below, shows the superiority of NB classifier in all tests. The figure illustrates well the results:

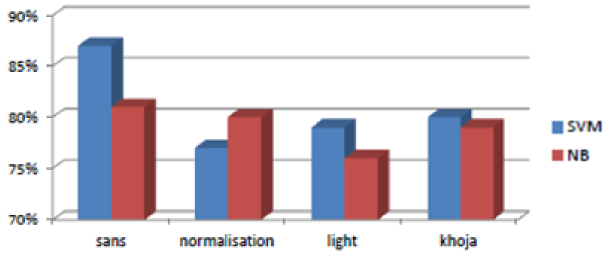

Fig. 3 NB, SVM comparaison (percentage split)

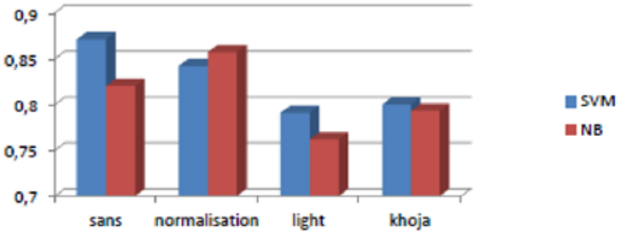

Fig. 4 NB, SVM precision comparaison (percentage split)

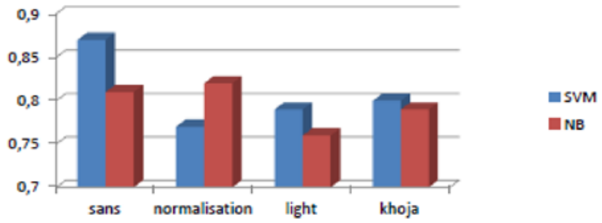

Fig. 5 NB, SVM recall comparaison (percentage split)

The application of light and stemming Khoja also degraded results. Here SVM was most effective with most pre-treatments, but it reached 0.871 precision and recall of 0.87 without any pretreatment with the estimation method of reliability "Percentage split."

The application of two classification algorithms, Naive Bayes (NB) and Support Vector Machines (SVM), on an Arabic corpus showed the superiority of the first over the second classifier. In the experiments we performed, we found that the naive Bayesian classifier gave the most successful results most of the time.

The results show the impact of the pretreatment phase and the application of different techniques with respect to a data set. The best results were achieved after the normalization of the corpus. The use of stemming and light stemming and Khoja stemming, degraded performance in most cases.

Despite its simplicity and the fact that the hypothesis of conditional independence obviously does not hold in real world situations, the NB classifier still tends to give effective results. Secondly, SVM has been shown to be very effective in the categorization of traditional text, usually surpassing NB.

5 Conclusion

This paper was primarily an exploration of the field of opinion analysis in Arabic. Throughout this work we encountered several challenges.

Among these challenges, we report the search for a reliable corpus tested by other similar research and explore a new area of research, namely natural language processing and especially Arabic.

We performed a binary classification (positive or negative) on 500 reviews of films. The two classifiers are used SVM and NB. This allowed us to compare the performance of two classifiers that are widely used in the classification field.

The experimental results showed the superiority of the classifier NB in most tests. The best result was obtained by performing a normalization of the corpus before the classification: this is our main contribution.

We also noted that the application of light stemming and Khoja stemming downgraded the results of our analysis of opinion.

References

1. About The Arabic language. World langages and culture. [En ligne] [Citation : 12 12 2015.] http://www.vistawide.com/arabic/arabic.htm.
2. About The Arabic language. (s.d.). Consulté le 12 12, 2015, sur World langages and culture: http://www.vistawide.com/arabic/arabic.htm
3. Mountassir, A., Benbrahim, H., & Berrada, I. (2012). SENTIMENT CLASSIFICATION ON ARABIC CORPORA: PRELIMINARY RESULTS OF A CROSSSTUDY. 3 e Séminaire de Veille Stratégique, Scientifique et Technologique (VSST'12).
4. Saad, M. K., & Ashour, W. (2010, November). Osac: Open source arabic corpora. In 6th ArchEng Int. Symposiums, EEECS (Vol. 10).

5. Al-Kabi, M., Al-Shawakfa, E., & Alsmadi, I. (2013). The Effect of Stemming on Arabic Text Classification: An Empirical Study. Information Retrieval Methods for Multidisciplinary Applications, 207.
6. Al-Shargabi, B., Al-Romimah, W., & Olayah, F. (2011, April). A comparative study for Arabic text classification algorithms based on stop words elimination. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications* (p. 11). ACM.
7. Alsaleem, S. (2011). Automated Arabic Text Categorization Using SVM and NB. Int. Arab J. e-Technol., 2(2), 124-128.
8. Hmeidi, I., Hawashin, B., & El-Qawasmeh, E. (2008). Performance of KNN and SVM classifiers on full word Arabic articles. *Advanced Engineering Informatics*, *22*(1), 106-111.
9. Moh'd Mesleh, A. (2011). Feature sub-set selection metrics for Arabic text classification. Pattern Recognition Letters, 32(14), 1922-1929.
10. El-Halees, A. (2008). A comparative study on Arabic text classification. Egyptian Computer Science Journal, 30(2).
11. Khorsheed, M. S., & Al-Thubaity, A. O. (2013). Comparative evaluation of text classification techniques using a large diverse Arabic dataset. Language resources and evaluation, 47(2), 513-538.
12. Heitz, T. (2006). Modélisation du prétraitement des textes. In JADT'06 (International Conference on Statistical Analysis of Textual Data) (Vol. 1, pp. 499-506).
13. Ayadi, R., Maraoui, M., & Zrigui, M. (2014). Latent Topic Model for Indexing Arabic Documents. International Journal of Information Retrieval Research (IJIRR), 4(1), 29-45.
14. Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.
15. Mounir Zrigui, Rami Ayadi, Mourad Mars, Mohsen Maraoui: Arabic Text Classification Framework Based on Latent Dirichlet Allocation. CIT 20(2): 125-140 (2012)
16. Mbarek Charhad, Mounir Zrigui, Georges Quénot : Une approche conceptuelle pour la modélisation et la structuration sémantique des documents vidéos, SETIT-3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, TUNISIA,(2005)
17. Mohamed Achraf Ben Mohamed, Sarra Zrigui, Anis Zouaghi, Mounir Zrigui: N-scheme model: An approach towards reducing Arabic language sparseness. ICTA 2015: 1-5
18. Mohamed Achraf Ben Mohamed, Souheyl Mallat, Mohamed Amine Nahdi, Mounir Zrigui: Exploring the potential of schemes in building NLP tools for arabic language. Int. Arab J. Inf. Technol. 12(6): 566-573 (2015)