

An Empirical Study on the effect of weighting schemes and Machine Learning algorithms on the Arabic text Classification

Rim Sakina BENNABI

EEDIS Laboratory Ecole Supérieure en Informatique Sidi Bel Abbes,
Algeria, bennabi.rim@gmail.com,

Zakaria ELBERRICHI

EEDIS Laboratory Djillali Liabes University UDL Sidi Bel Abbes,
Algeria, elberrichi@gmail.com,

Abstract—Nowadays, many applications use text classification to categorize different documents with predefined labels. However, most of the work focuses on the English language and a small number of studies focus on the Arabic language. The latter is widely used on the Internet and has a complex morphology that needs to be studied in different manners. In this context, we propose in this paper an empirical study on the effect of the use of different learning algorithms (SVM, Naive Bayes and KNN) and different weighting methods (TC, TF and TF.IDF) on Arabic textual classification. The goal of our work is to find the best combination that enhance the performance. The results show that SVM and TF.IDF combination offers the best accuracy and F-Measure (94%).

Index Terms—weighting schemes, SVM, NB, KNN, Arabic text classification.

I. INTRODUCTION

Textual classification is a process that primarily aims to categorize documents into specific classes. It is a supervised approach that lies at the intersection of three different disciplines: machine learning, automatic natural language processing (NLP) and information retrieval. A lot of work is currently, focused on English texts: it is the dominant language of the Web. However, a need arises for the other languages, because the Web is each day more multilingual[1]. Arabic is a Semitic language of rich and complex morphological structure, it requires on the one hand several investigative works in order to improve the performances of the textual classification techniques applied to the documents in Arabic at any stage (from the preprocessing to the Evaluation). On the other hand, unlike the English language and other Latin languages where corpora and Benchmarks are widely available, the language of “DADE” suffers from an enormous lack of open corpora, which also requires to collect documents and to provide efforts in order to build labeled and unlabeled Arabic datasets.

Also, most research work done in this context and in the supervised classification area usually involves the implementation of a very specific technique or algorithm. But combinatorial methods are known to perform effectively and to provide better results, so the experimentation with Arabic texts seems to be an interesting research field. In this work, we study different combination between ML algorithms and weighting schemes for the Arabic document categorization.

The rest of the paper is organized as follows : Section 2 introduces the Arabic language and its characteristics. Section 3 discuss the text mining process. Section 4 presents the related works. Experimental results and discussions are explained in section 5, and finally conclusion of this work is presented in section 6.

II. ARABIC LANGUAGE

A. Presentation of the Arabic Language

Among the most widely used languages, the Arabic language ranks the fifth one [2], and ranks the fourth in the ranking list of the ten most used languages on the Internet with 185 million users, and this according to a statistical study done on the site (<http://www.internetworldstats.com>) the month of June 2017.

This is a language that is written with 28 letters from right to left in addition to Hamza (ء). All letters are consonants except (ي و ا) considered as vowels. It has several rules regarding inflections, gender and number as well as complex morphology.

B. Characteristics of the Arabic Language

Diacritics

The presence of diacritics helps to limit the ambiguity of pronunciation and the meaning of words. The main diacritics that can not be removed from a text in Arabic and that are mandatory are the point and Hamza. The point is used to distinguish between different letters written in the same way, example : ع and غ. Hamza can act as a letter or as diacritical.

Other diacritics facilitate reading and are essentially divided into two classes: vocalization signs and syllabic signs. Vocalization signs include: Short vowels (Fatha, Damma, Kasra), long vowels (المدة), Tanween (التنوين), Alif madda (ألف المدة), Sukun and Šadda (شدة).

Names, verbs and articles:

The Arabic language has two genres (male, female), three states (singular, double, plural) and three grammatical states (2) [الرفع النصب الجر]. A word in Arabic can be either a verb (فعل), a noun (اسم) or an article (3) [حرف].

Prefixes, Suffixes and Infixes:

A prefix is a set of letters that precede the root word, while a suffix succeeds the root. The Arabic language also has infixes (letters that are mixed with the root inside the word) [4].

III. TEXT CLASSIFICATION PROCESS

Text classification or text categorization is a supervised method that assigns a document to a predefined class (simple classification) or set of classes (multi-classification). Formally, the textual classification is expressed by the function (classifier):

$$\varphi: D \times C \rightarrow (T, F)(1)$$

Where D is the set of documents and C is the set of predefined categories.

There are three main steps in a classification system:

- Preprocessing step : it includes the preparation of dataset and features selection to represent the document according to the weighting scheme.

- Model Extraction step : It consists of the machine learning algorithm application to extract the classification model.
- The evaluation step : In this stage the classification model is applied to the test set to evaluate the system.

A. Preprocessing

Before transforming textual data into a structured form, a number of preprocessing methods can be used, including text cleaning, tokenization, filtering, lemmatisation, stemming and other linguistic preprocessing.

The text has to be transformed later as a vector space model. This model represents documents as vectors of attributes in an m-dimensional space; m being the number of distinct tokens (n-grams, words, sentences, ...) of the corpus used. The document is then represented by a weight vector of tokens. The weighting approaches are diverse, the most important ones are:

- The binary approach: expresses the presence or absence of a term in a document.
- The frequency of the term (TC): number of occurrences.
- The relative frequency(TF): number of occurrences by number of tokens in the document.
- TF*IDF : TF is the frequency of the term, and the IDF of the term t is calculated by :

$$IDF(t) = \log \left(\frac{N}{df(t)} \right) \quad (2)$$

Where : N is the number of documents, and df(t) is the number of documents containing the term t.

B. Model extraction

For the classification step, there are many machine learning algorithms and methods, We only present those used in our experiments.

The NB algorithm

Naive Bayes algorithm is a probabilistic algorithm using the Bayes' theorem, it assumes that all the words of a document d are generated by a probabilistic mechanism, and that the class of a document is linked to the words that appear with this class. Since $P(C/d)$ is the probability that a document d belongs to category C, and by applying the Bayes theorem, we obtain:

$$P(C/d) = \frac{P(d/C) \times P(C)}{P(d)} \quad (3)$$

The document d is assigned to the class with the highest posterior probability :

$$C_d = \operatorname{argmax}(P(C_i/d)); 1 \leq i \leq |C| \quad (4)$$

The Naive call comes from the fact that the terms are supposed to be independent . This assumption is not true. However, naïve bayes algorithm is widely used and presents satisfactory results.

KNN algorithm

The KNN algorithm is a very simple supervised classification method that consists of electing the nearest neighbors of a new document based on a distance function. The class of d is determined , for example, by the majority class among these k nearest neighbors. KNN is a lazy algorithm, it means that learning phase or model construction is not required.

The KNN algorithm is based on the use of similarity measures. The widely used measures are Euclidean distance, Manhattan distance and cosine-based similarity.

SVM algorithm

The Support Vector Machines (SVM) is an algorithm known as fast and efficient for classification problems. It is based on the notion of maximum margin which is the distance between the separation boundary and the closest samples (support vectors). For example, a binary SVM algorithm can be seen as a hyper-plane in a space separating points representing positive instances from one representing negative instances. The separation boundary that maximizes the margin is selected.

C. Evaluation

A part of the corpus called test set is used for the evaluation phase. The documents of this set are classified with the extracted model obtained during the learning phase and the predicted labels are compared with the real ones. The common evaluation measures for classification are: Accuracy, precision, recall and F-Measure.

IV. RELATED WORKS

[5] performs a textual categorization of documents by integrating the Arabic morphological analysis tool Khoja Stemmer with the well-known WEKA machine learning software, by applying the REP-Tree classification algorithm. Two main combinations were used, the first one is the bag-of-words (BoW) without application of stemming, the second one is the combination of terms with stemming (Khoja Stemmer). The experiments with these two representations were applied to the BBC corpus (comprising 110 documents and 7 categories) and repeated for different weight schemes (TF, IDF, TF / IDF, Norm, minFreq3, NormminFreq3). same experiments were performed with use of G-graph as learning algorithm instead of REP-Tree.

The research in [6] applies three classification algorithm on the Arabic data set : KNN, C4.5 and Rocchio. These well-known algorithms are applied on in-house collected Arabic data set. The corpus consists from 1400 documents belongs to 8 categories. The experimentation results show that precision and recall results using Rocchio classifier and K-NN are better than those using C4.5 algorithm. Also, the results - using different variations of k parameter of the K-NN classifier – give bad results for almost all types of categories when k=1 . However, the results are promising when k varies between 12 and 18.

In [7], a system named ArabCat has been proposed for Arabic textual classification making use of maximal entropy. This system uses Aljazeera News's collected corpus which is given as input to the preprocessor module that normalizes data by removing punctuation, non-alphabetic characters and converting text into UTF-8 encoding. Then the documents are processed by the "parser" and all the empty words are eliminated, this phase is followed by a stems extraction and in a number of experiments the POS labeling is also applied. The second module of the system the "trainer" supports the learning part of the documents and applies the maximum entropy algorithm to extract the model. The third module "classifier" takes as input preprocessed unclassified documents and assigns a category for each document based on the model provided by the "trainer".

The work of [8] presents an application of Artificial Neural Networks (ANN) for the documents classification in Arabic using a combination of dimensionality reduction methods. The neural network used is contains three fully connected layers. Neurons are nonlinear units with hyperbolic tangent (tanh). The activation function in the output layer is the sigmoid function and the learning of the neural network is based on the Back Propagation algorithm. The number of input data is equal to the dimensionality of the reduced feature space and the output di-

mension to the number of categories. The selection of the characteristics is carried out by combining the TF.IDF method with the DF method and the CF method, and this by selecting only the terms with the highest TF / IDF weightings and exceeding the minimum threshold DF (document frequency) and the minimum threshold CF (class frequency).

V. EMPIRICAL STUDY

A. The Data Set :

The used data sets are the Khaleej-2004 corpus and the Watan Corpus.

- Khaleej-2004 corpus [9] contains 5690 documents divided into four categories Economy, International news, Local news and Sports.
- Watan corpus[10] contains 20291 documents divided into 6 categories culture, economy, international, local, religion and sports.

The corpus were evaluated using percentage split method (holdout), where 70% of the data were used as training instances and the remaining (30%) were used as testing instances (Table I).

TABLE I. DOCUMENTS NUMBER PER SET

	Khaleej-2004	Watan-2004
Number of documents	5690	20291
Number of documents in training set	3812	13594
Number of documents in test set	1878	6697

B. Text classification process:

In our study, we perform multiple combinations of the text mining process to extract the most efficient one.

Preprocessing

The stage of preprocessing includes only the punctuation suppression, the tokenisation of the words and the transformation to vectors using three different ponderation schemes : TC (number of term occurrences), TF(term frequency) and TF.IDF.

Machine Learning Algorithms

Three machine learning algorithms are used for the classification tasks: Naive Bayes – KNN and SVM. (See the model extraction in section 3.

Evaluation Measures

For the evaluation, we used the well-known classification measures : Accuracy, Precision, recall and F-Measure.

C. Obtained Results

Experimentations with Khaleej-2004 + TC weighting scheme

TABLE II. KHALEEJ-2004 + TC RESULTS

	Accuracy	Precision	Recall	F-Measure
Naive Bayes	93.98 %	0.94	0.94	0.94
SVM	93.50 %	0.93	0.94	0.93
KNN	76.99 %	0.80	0.77	0.77

Using the TC for term weighting, Naive Bayes generates the best F-measure (94%), SVM also shows a good performance (F-measure = 93%).

Experimentations with Khaleej-2004 + TF weighting scheme

TABLE III. KHALEEJ-2004 + TF RESULTS

	Accuracy	Precision	Recall	F-Measure
Naive Bayes	67.89 %	0.82	0.68	0.73
SVM	93.18 %	0.93	0.93	0.93
KNN	84.13 %	0.85	0.84	0.84

Using the TF as term weighting, the SVM algorithm outperformed the other two algorithms, having the same F-measure as the previous experiment. KNN improved its F-Measure, however NB achieved a lower performance than TC.

Experimentations with Khaleej-2004 + TF.IDF weighting scheme

TABLE IV. KHALEEJ-2004 + TF.IDF RESULTS

	Accuracy	Precision	Recall	F-Measure
Naive Bayes	81.84 %	0.87	0.82	0.80
SVM	94.03 %	0.94	0.94	0.94
KNN	92.43 %	0.92	0.92	0.92

With the NB as a classification algorithm, the best performances remain those obtained with the TC weighting, however the TF.IDF is better than the TF. The SVM algorithm is rather stable in its results and KNN had the best F-measure with this experiment.

These three experiments were repeated, this time using the Watan-2004 corpus (which is larger than Khaleej-2004) as a data set, to verify the scalability of the different approaches.

Experimentations with Watan-2004 + TC weighting scheme

TABLE V. WATAN-2004 + TC RESULTS

	Accuracy	Precision	Recall	F-Measure
Naive Bayes	90.78 %	0.91	0.91	0.91
SVM	84.99 %	0.88	0.85	0.85
KNN	71.21 %	0.73	0.71	0.71

By increasing the size of the corpus, the results obtained with the TC weighting decreased in accuracy and F-measure for all the used classification algorithms.

Experimentations with Watan-2004 + TF weighting scheme

TABLE VI. WATAN-2004 + TF RESULTS

	Accuracy	Precision	Recall	F-Measure
Naive Bayes	80 %	0.84	0.80	0.81
SVM	90.05 %	0.90	0.90	0.90
KNN	81.26 %	0.82	0.81	0.81

Experimentations with Watan-2004 + TF.IDF weighting scheme

TABLE VII. WATAN-2004 + TF.IDF RESULTS

	Accuracy	Precision	Recall	F-Measure
Naive Bayes	86.51 %	0.88	0.87	0.86
SVM	91.99 %	0.92	0.92	0.92
KNN	88.51 %	0.89	0.89	0.89

The last two experiments showed that by increasing the size of the corpus, the results obtained with the TF and the TF.IDF weighting decreased in performance, for the KNN and SVM

algorithms. However, NB has increased in F-Measure and accuracy.

D. Summary of results according to accuracy

Khaleej-2004 Results

TABLE VIII. SUMMARY OF KHALEEJ-2004 ACCURACY RESULTS

	TC	TF	TF.IDF
Naive Bayes	93.98 %	67.89 %	81.84 %
SVM	93.50 %	93.18 %	94.03 %
KNN	76.99 %	84.13 %	92.43 %

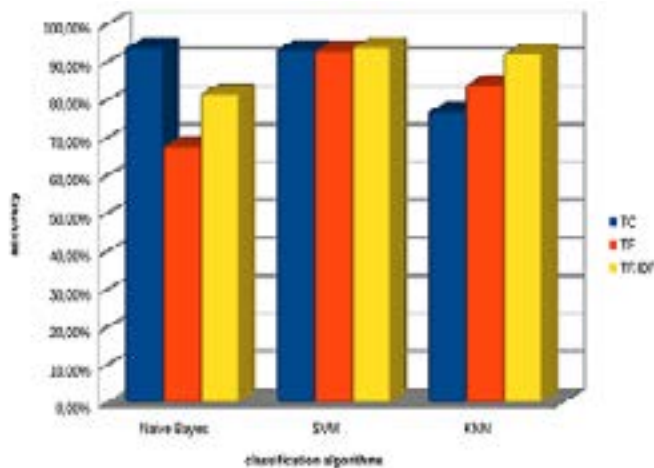


Fig.1. Khaleej-2004 : Effects of weighting schemes on classification accuracy

Watan-2004 Results

TABLE IX. SUMMARY OF WATAN-2004 ACCURACY RESULTS

	TC	TF	TF.IDF
Naive Bayes	90.78 %	80 %	86.51 %
SVM	84.99 %	90.05 %	91.99 %
KNN	71.21 %	81.26 %	88.51 %

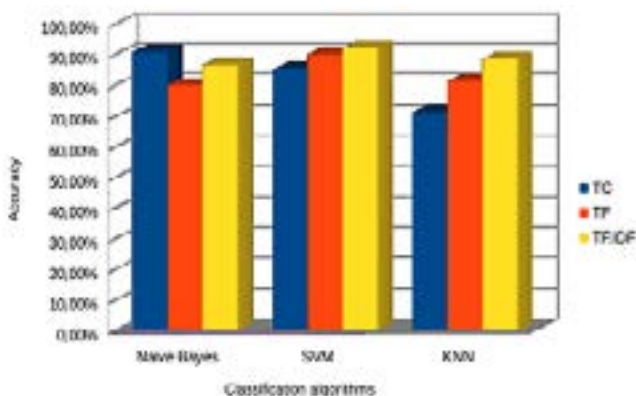


Fig.2. Watan-2004 : Effects of weighting schemes on classification accuracy

E. Results Interpretation

The first point to note is that the results obtained with the various evaluation measures are consistent (for example, an increase in accuracy corresponds to an increase in the F-Measure and vice versa).

Also, the results obtained with the Khaleej-2004 corpus are slightly better than those obtained with Watan-2004 (a difference of approximately 3%) except for Naive Bayes with TF and

TF.IDF weighting schemes.

The best performances obtained with the two corpora are those obtained by using the SVM combined with the TF.IDF weighting followed by the Naive Bayes combined with the TC weighting. From these results, we deduce that some combinations offer better performances than others (SVM + TF.IDF, NB + TC) and that the machine learning algorithms SVM and Naive Bayes known to be effective for textual classification in English are also efficient for the Arabic language. on the other hand, we note that the increase in the size of the corpus affects the results.

VI. CONCLUSION

In this experimental study, we compared the different classification methods including Naive Bayes, SVM and KNN as well as the different weighting schemes used when converting text into document vectors (TC, TF and TF.IDF). These experiments have shown that some combinations offer better performance for textual classification in Arabic and that other combinations seem to be well suited for scaling up. On the basis of this work, we will start other research experiments that may be promising for textual classification in Arabic such as the use of different pretreatment methods (Ngrams, concepts, stems, ...), the use of ensemble methods or the use of meta-heuristics.

REFERENCES

- [1] Elberichi, Z., and Abidi, K. "Arabic Text Categorization: a Comparative Study of Different Representation Modes", International Arab Journal of Information Technology, 2012, 9, (5)
- [2] Saad M., "The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification," Master Thesis, The Islamic University, Gaza, 2010.
- [3] Khoja, S. "APT: Arabic part-of-speech tagger", Proc. Proceedings of the Student Workshop at NAACL 2001.
- [4] Al-Shammari, E., and Jessica L. "A novel Arabic lemmatization algorithm." Proceedings of the second workshop on Analytics for noisy unstructured text data, ACM, 2008.
- [5] Naji, H., and Wesam A. "Text Classification for Arabic Words Using Rep-Tree." International Journal of Computer Science & Information Technology (IJCSIT) Vol 8, No 2, April 2016.
- [6] Mohammad, A.H., Al-Momani, O., and Alwada'n, T.: "Arabic Text Categorization using k-nearest neighbour, Decision Trees (C4. 5) and Rocchio Classifier: A Comparative Study", 2016.
- [7] El-Halees, A. M. "Arabic text classification using maximum entropy", IUG Journal of Natural Studies", 2015, 15(1).
- [8] Al Zaghoul, F., and Al-Dhaheri, S.: 'Arabic Text Classification Based on Features Reduction Using Artificial Neural Networks' (Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on IEEE, 2013.
- [9] Abbas, M., and K. Smaili. "Comparison of Topic Identification Methods for Arabic Language", in RANLP05: Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2005, 14-17.
- [10] Abbas, M., Smaili K., and Berkani D. "Evaluation of Topic Identification Methods on Arabic Corpora", Journal Of Digital Information Management, 2011, 9(5): 8.