I.J. Intelligent Systems and Applications, **2017, 4, 39-52**
Published Online April 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2017.04.05

Combining Different Approaches to Improve Arabic Text Documents Classification

Ibrahim S. I. Abuhaiba
Computer Engineering Department, Islamic University, P. O. Box 108, Gaza, Palestine
E-mail: isiabuhaiba@gmail.com

Hassan M. Dawoud
Computer Engineering Department, Islamic University, P. O. Box 108, Gaza, Palestine

***Abstract*—**The objective of this research is to improve Arabic text documents classification by combining different classification algorithms. To achieve this objective we build four models using different combination methods.

The first combined model is built using fixed combination rules, where five rules are used; and for each rule we used different number of classifiers. The best classification accuracy, 95.3%, is achieved using majority voting rule with seven classifiers, and the time required to build the model is 836 seconds.

The second combination approach is stacking, which consists of two stages of classification. The first stage is performed by base classifiers, and the second by a meta classifier. In our experiments, we used different numbers of base classifiers and two different meta classifiers: Naïve Bayes and linear regression. Stacking achieved a very high classification accuracy, 99.2% and 99.4%, using Naïve Bayes and linear regression as meta classifiers, respectively. Stacking needed a long time to build the models, which is 1963 seconds using naïve Bayes and 3718 seconds using linear regression, since it consists of two stages of learning.

The third model uses AdaBoost to boost a C4.5 classifier with different number of iterations. Boosting improves the classification accuracy of the C4.5 classifier; 95.3%, using 5 iterations, and needs 1175 seconds to build the model, while the accuracy is 99.5% using 10 iterations and requires 1966 seconds to build the model.

The fourth model uses bagging with decision tree. The accuracy is 93.7% achieved in 296 seconds when using 5 iterations, and 99.4% when using 10 iteration requiring 471 seconds. We used three datasets to test the combined models: BBC Arabic, CNN Arabic, and OSAC datasets. The experiments are performed using Weka and RapidMiner data mining tools. We used a platform of Intel Core i3 of 2.2 GHz CPU with 4GB RAM.

The results of all models showed that combining classifiers can effectively improve the accuracy of Arabic text documents classification.

***Index Terms*—**Text classification, combining classifiers, fixed combining rules, stacking, boosting, bagging.

I. Introduction

Text classification is a technique often used as a basis for applications in document processing, Web mining, topic identification, text filtering, document organization, etc. Many methods and algorithms that vary in their accuracy have been applied to the problem of text classification. Assessment of different methods by experiment is the basis for choosing a classifier as a solution to a particular problem instance. There are several methods used to classify text such as Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Artificial Neural Networks (ANN), Naïve Bayes Classifier (NB), and Decision Trees (DT).

Often none of the basic traditional single classifiers is powerful enough to distinguish the pattern classes optimally. For more complicated datasets, the traditional set of classifiers can be improved by various types of combining rules [1]. Therefore, we need an effective methodology for combining them. There are three main motivations to combine classifiers [2]:

Statistical motivation: it is possible to avoid the worst classifier by averaging several classifiers, which was confirmed theoretically [3], and was demonstrated to be efficient in many applications.

Representational motivation: under particular situations, fusion of multiple classifiers can improve the performance of the best individual classifier. It happens when the optimal classifier for a problem is outside the considered classifier space.

Computational motivation: some algorithms suffer from local minima and perform an optimization task in order to learn. Algorithms such as the back propagation for neural networks are initialized randomly in order to avoid local optimum solutions. In this case, it is a difficult task to find the best classifier, and it is often used several (hundreds or even thousands) initializations in order to find a presumable optimal classifier. Combination of such classifiers showed to stabilize and improve the best single classifier result.

A. Combining Classifiers

The general idea of combining classifiers can be summarized by the use of a methodology to create an ensemble of learners and to produce a final decision

given the outputs of those learners. This kind of models is intuitive since it imitates our nature to seek several opinions before making a crucial decision [4].

In this paper, we will use four models to combine classifiers to improve the classification of Arabic text documents. These models are fixed combining rules, stacking, AdaBoost, and bagging.

In the fixed combining rules, all classifiers are trained and each classifier gives its decision, then the combiner uses the results of classifiers to give the final decision according to the rule used for combination. Many rules can be used in the combiner such as majority voting, maximum rule, minimum rule, average rule, and product rule [5, 6].

Stacking is probably the most popular meta-learning technique [7]. It is usually employed to combine models built by different classifiers. The stacking algorithm is based on two levels of classification. The first level contains the base classifiers that are trained using the original dataset. Then, a new dataset is generated using the original dataset and the prediction of base classifiers. This dataset is used to train the meta classifier that combines the different predictions into a final one [4].

AdaBoost tries to combine weak base classifiers in order to produce an accurate strong classifier [8]. The approach is an iterative process that builds an ensemble of classifiers. The algorithm trains the classifier sequentially, a new model per round. At the end of each round, the misclassified patterns are weighted in order to be considered more important in the next round, so that the subsequent models compensate error made by earlier classifiers. The learning algorithm of the classifier used in AdaBoost must allow the use of a weight for each training pattern. The idea is to give higher weights to the patterns that are misclassified and in the next iteration try to construct a classifier capable of classifying correctly these kinds of patterns [4].

The bagging technique (bootstrap aggregating) [9] is based on the idea that bootstrap samples of the original training set will present a small change with respect to the original training set, but sufficient difference to produce diverse classifiers. Each member of the ensemble is trained using a different training set, and the predictions are combined by averaging or voting. The different datasets are generated by sampling from the original set, choosing N items uniformly at random with replacement [5].

B. Arabic Language

Arabic language is one of the widely used languages in the world. Arabic language is a Semitic language that has a complex and much morphology than English; it is a highly inflected language [10].

Arabic language consists mainly of 28 alphabet characters. Arabic is written from right to left. Arabic letters have different styles when appearing in a word depending on the letter position at beginning, middle, or end of a word and on whether the letter can be connected to its neighbor letters or not [11].

Arabic words have two genders: feminine and masculine; three numbers: singular, dual, and plural; and three grammatical cases: nominative, accusative, and genitive. A noun has the nominative case when it is subject; accusative when it is the object of a verb; and the genitive when it is the object of a preposition. Words are classified into three main parts of speech: nouns (including adjectives and adverbs), verbs, and particles. All verbs and some nouns are morphologically derived from a list of roots. Words are formed by following fixed patterns; the prefixes and suffixes are added to the word to indicate its number, gender, and tense [11].

Arabic language is a challenging language for a number of reasons [12]:

- It is orthographic with diacritics. So, it is less ambiguous and more phonetic. Certain combinations of characters can be written in different ways.
- Arabic has a very complex morphology recording as compared to English language.
- Arabic words are usually derived from a root (a simple bare verb form) that usually contains three letters. In some derivations, one or more of the root letters may be dropped. In such cases, tracing the root of the derived word would be a much more difficult problem.
- Broken plurals are common. Broken plurals are somewhat like irregular English plurals except that they often do not resemble the singular form as closely as irregular plurals resemble the singular in English. Because broken plurals do not obey normal morphological rules, they are not handled by existing stemmers.
- In Arabic, we have short vowels that give different pronunciation. Grammatically, they are required but omitted in written Arabic texts.
- Arabic synonyms are widespread. Arabic is considered one of the richest languages in the world. This makes exact keyword match is inadequate for Arabic retrieval and classification

There are many researches for classification of text using different classification techniques. These researches are mainly applied to English documents, but in Arabic it is still limited [10, 11, 13, 14, 15]. Previous researchers applied single classifiers to classify Arabic documents, but in this paper we will combine multiple classifiers aiming to a more accurate classification.

The rest of this paper is organized as follows. Section II presents related work. In Section III, our methodology is described. In Section IV, experimental results of our work are presented, discussed, analyzed, and compared with different single classifiers that have been applied to Arabic text documents. Finally, the paper is concluded in Section V.

II. Related Work

Many researchers have worked on text classification in

English and other European languages. However, researches on text classification for Arabic language are limited [10, 11, 13, 14, 15].

In [16], SVMs are applied to classify Arabic articles with Chi Square feature selection in the pre-processing step. The reported F-measure is 88.1%. The author compared six feature selection methods with SVMs. He concluded that Chi Square method is the best. He used an in-house collected corpus from online Arabic newspaper archives, including Al-Jazeera, Al-Nahar, Al-hayat, Al-Ahram, and Al-Dostor as well as a few other specialized websites. The collected corpus contains 1445 documents that vary in length. These documents fall into nine classification categories (computer, economics, education, engineering, law, medicine, politics, religion, and sports) that vary in the number of documents. In the pre-processing step, each article in the dataset is processed to remove the digits and punctuation marks. He applied normalization of some Arabic letters. In addition, all non-Arabic text is filtered, and he did not apply stemming.

In [17], the authors applied neural networks (NN) to classify Arabic text. Their experimental results show that using NN with Singular Value Decomposition (SVD) as a feature selection technique gives better result, 88.3% accuracy, than the basic NN (without SVD), 85.7% accuracy. They also experienced the scalability problem with high dimensional text dataset using NN. They collected the corpus from Hadith encyclopedia from the nine books. It contains 435 documents belonging to 14 categories. They applied light stemming and stop words removal on the corpus. Term Frequency-Inverse Document Frequency (TF-IDF) is used as a weighting scheme.

In [18], the authors classified Arabic text documents using Naïve Bayes classifier (NB). The average accuracy is 68.8%, and the best accuracy is 92.8%. They used a corpus of 1500 text documents belonging to five categories; each category contains 300 text documents. All words in the documents are converted to their roots. The vocabulary size of resultant corpus is 2,000 terms/roots. Cross-validation is used for evaluation.

Maximum entropy is used in [19] for Arabic text classification, and in [20] to classify and cluster news articles. The best classification accuracy reported in [19] is 80.4% and 62.7% in [20].

kNN has been applied in [21] for Arabic text classification. They used TF-IDF as a weighting scheme and got an accuracy of 95%. They also applied stemming and feature selection. The authors reported the problem of lacking freely public availability of Arabic corpus. They collected a corpus from newspapers (Al-Jazeera, An-Nahar, Al-Hayat, Al-Ahram, and Ad-Dostor) and from Arabic Agriculture Organization website. The corpus consists of 621 documents belonging to 1 of 6 categories (politics 111, economic 179, sport 96, health and medicine 114, health and cancer 27, agriculture 100). They preprocessed the corpus by applying stop words removal and light stemming.

In [22], the authors compared kNN and SVM for Arabic text classification. They used full word features and considered TF-IDF as the weighting method for feature selection, and Chi statistics for ranking metrics. They showed that both SVM and kNN have superior performance, and SVM has better accuracy and time. Authors collected documents from online newspaper (Al-Ra'i and Ad-Dostor). They collected 2206 documents for training and 29 documents for testing. The collected documents belong to one of two categories (sport and economic).

In [23], the authors compared Triggers Classifier (TR-Classifier) and kNN to identify Arabic topics. kNN uses the whole vocabulary (800), while TR uses reduced vocabulary (300). The average recall and precision for kNN and TR are 0.75, 0.70 and 0.89, 0.86, respectively. They collected 9,000 articles from Omani newspaper (Al-Watan). The corpus belongs to 1 of 6 categories (culture, economic, religious, local news, and international news). The corpus includes 10M words including stop words. After removing stop and infrequent words, the vocabulary size became 7M words. TF-IDF was used as a weighting scheme.

In [11], three popular text classification algorithms are compared (kNN, NB, and Distance-Based classifier). Experimental results show that NB outperforms the other two algorithms. 1,000 text documents were collected belonging to 10 categories (sport, economic, internet, art, animals, technology, plants, religious, politics, and medicine). Each category contains 100 documents. The corpus was preprocessed by applying stop words removal and stemming. One-half of the documents was used for training and the other half for testing.

In [15], three classification algorithms are compared to classify Arabic text: kNN, NB, and Rocchio. NB was the best performing algorithm. The author collected the corpus from online newspapers (Al-Jazeera, An-Nahar, Al-Hayat, Al-Ahram, and Ad-Dostor). The corpus consists of 1,445 documents belonging to nine categories (medicine 232, sport 232, religious 227, economic 220, politics 184, engineering 115, low 97, computer 70, and education 68). They applied light stemming for feature reduction. Cross-validation was performed for evaluation.

In [10], the authors evaluated the performance of two popular text classification algorithms (SVMs and C5.0) to classify Arabic text using seven Arabic corpora. The average accuracy achieved is 68.7% and 78.4% by SVMs and C5.0, respectively. One of the goals of their paper is to compile Arabic corpora to be benchmark corpora. The authors compiled seven corpora consisting of 17,658 documents and 11,500,000 words including stop words. The corpora are not available publically.

In [24], the authors applied kNN and NB on Arabic text and concluded that kNN has better performance than NB; they also concluded that feature selection, the size of training set, and the value of k affect the performance of classification. The researchers also posed the problem of unavailability of freely accessible Arabic corpus. The in-house collected corpus consists of 242 documents belonging to six categories. Authors applied light stemming as a feature reduction technique and TF-IDF as weighting scheme; they also performed a cross-validation

test.

In [14], the author compared six well know classifiers applied on Arabic text: ANN, SVM, NB, kNN, Maximum Entropy, and Decision Tree. He showed that NB and SVMs are the best classifiers in terms of F-Measure with values of 91% and 88%, respectively. He also applied information gain in feature selection. The reported F-Measure was 83% and 88% for NB and SVMs, respectively. He collected Arabic documents from the Internet, mainly from Aljazeera Arabic news channel. The documents are categorized into six domains: politics, sports, culture and arts, science and technology, economy, and health. The author applied stop words removal and normalization and used 10-folds cross-validation for testing.

In [13], the author compares the impact of text preprocessing on Arabic text classification using popular text classification algorithms: Decision Tree, k Nearest Neighbors, Support Vector Machines, Naïve Bayes and its variations. He applied different term weighting schemes, and Arabic morphological analysis (stemming and light stemming). He used seven Arabic corpora (3 in-house collected and 4 existing corpora). The experiments showed that light stemming with term pruning is the best feature reduction technique, which reduced features to 50% of the original feature space. Support vector machines and Naïve Bayes variations achieved the best classification accuracy and outperformed other algorithms. Weighting schemes impact the performance of distance based classifiers.

Many researches showed that combining classifiers can enhance the results of classification in general, but combining classifiers is not widely used to classify Arabic documents. In [25], the author presented a combined approach consisting of three methods that automatically extracts opinions from Arabic documents. At the beginning, a lexicon-based method is used to classify as much documents as possible. The resultant classified documents are used as a training set for a maximum entropy method that subsequently classifies some other documents. Finally, k-Nearest Neighbor method used the classified documents from the lexicon based method and maximum entropy method as a training set and classifies the rest of the documents. Experiments showed that, in average, the accuracy moved from 50% when using only lexicon based method to 60% when used lexicon based method and maximum entropy together, and to 80% when using the three combined methods.

In [26], documents are represented as vectors where each component is associated with a particular word. The authors propose voting methods, ordered weighted averaging (OWA) operator, and Decision Template method for combining classifiers. Experimental results showed that these methods decrease the classification error to 15% as measured on 2000 documents of training data from 20 newsgroups dataset.

In [27], the authors provide good statistical classifiers with generalization ability for multi-label categorization and present a classifier design method based on approach combination and F1-score maximization. They design multiple models for binary classification per category, and then combine these models to maximize the F1-score of a training dataset. Experimental results confirmed that the method is useful especially for datasets where there are many combinations of category labels.

In [28], the authors present an investigation into the combination of four different classification methods for text categorization using Dempster's rule of combination. These methods include the SVM, kNN, kNN model-based approach (kNNM), and Rocchio methods. They present an approach for effectively combining the different classification methods. Then, they apply these methods to a benchmark data collection of 20-newsgroups, individually and in combination. Experimental results show that the performance of the best combination of the different classifiers on 10 groups of the benchmark data can achieve 91.1% classification accuracy, which is 2.7% better than SVM.

III. Methodology

To implement and evaluate our approaches we use the following steps:

1. Collecting data: collect Arabic text documents from different domains.
2. Preprocessing data: through applying different text pre-processing techniques, which include applying a term, weighting scheme, and Arabic morphological analysis (stemming and light stemming).
3. Combining classifiers: by combining different classification algorithms and using different combining techniques.
4. Model evaluation: we use accuracy, precision, recall, and F-Measure.
5. Compare our results of combined classifiers with other results using single classifiers.

Table 1. CNN Arabic Corpus

Category	Number of Documents
Business	836
Entertainments	474
Middle East News	1462
Science & Technology	526
Sports	762
World News	1010
Total	5070

We use freely public Arabic datasets [29]. The first dataset was collected from CNN Arabic website. Table 1 presents domains of CNN-Arabic corpus that includes 5070 documents. Each document belongs to 1 of 6 categories. The second dataset to be used is called OSAC that was collected from multiple websites. The corpus includes 22,429 text documents. Each text document belongs to 1 of 10 categories as shown in Table 2. The third dataset was collected from BBC Arabic website. Table 3 presents domains of BBC-Arabic corpus which

includes 4,763 documents. Each document belongs to 1 of 7 categories.

Table 2. OSAC Dataset

Category	Number of Documents
Economic	3102
History	3233
Education and Family	3608
Religious and Fatwas	3171
Sport	2419
Health	2296
Astronomy	557
Low	944
Stories	726
Cooking Recipes	2373
Total	22,429

Table 3. BBC Arabic Corpus

Category	Number of Documents
Middle East News	2356
World News	1489
Business	296
Science & Technology	232
Sports	219
Entertainments	122
World Press	49
Total	4,763

Text preprocessing includes the following steps: tokenizing the document into words, normalizing tokenized words, stop word removal, stemming, and finally term weighting.

Tokenization is the task of separating running text into units. These units could be characters, words, numbers, sentences, or any other appropriate unit [30]. The definition of a word here is not the exact syntactic form, which is why we call it a token. In the case of Arabic, where a single word can be comprised of up to four independent tokens, morphological knowledge is needed to be incorporated into the tokenizer. One of the most useful features in detecting boundaries of sentences and tokens is punctuation marks. The number of punctuation marks and symbols used in Arabic corpus is 134 [31]. There are several methods to implement tokenization; the simplest way we used is extracting any alphanumeric string between two white spaces.

Normalization is the process of unification of different forms of the same letter. The corpus is normalized by the following steps: punctuations removal, diacritics removal, non-letter removal, replacing of إ ,أ, and آ with ا, replacing final ى with ي, and replacing final ة with ه.

Stop words generally carry no information. They are filtered out prior to processing [32].

Two major approaches are followed for Arabic stemming. One approach is called light stemming (also called stem-based stemming) where word's prefixes and suffixes are removed. The other one is called root-based stemming (also called aggressive stemming) which reduces a word to its root. Two other approaches are statistical stemming and manual construction of dictionaries; the last one is not efficient. Studies showed that light stemming outperforms aggressive stemming and other stemming approaches [33].

Arabic words are formed from abstract forms named roots, where the root is the basic form of a word from which many derivations can be obtained by attaching certain affixes producing many nouns, verbs, and adjectives from the same root [34]. A root-based stemmer main goal is to extract the basic form for any given word by performing morphological analysis on the word [35]. Khoja stemmer [36] basically attempts to find roots for Arabic words that are far more abstract than stems. It first removes prefixes and suffixes, then attempts to find the root for the stripped form. The problem in this stemming technique is that many different word forms are derived from an identical root, and so the root extraction stemmer creates invalid conflation classes that result in an ambiguous query, which leads to a poor performance [37].

Light stemming is to find a representative indexing form of a word by the application of truncation of affixes [38]. The main goal of light stemming is to retain the word meaning intact and so improves the retrieval performance of an Arabic information retrieval system. Many light stemming methods as Leah [39] stemmer classifies the affixes that can be attached to words to four kinds: antefixes, prefixes, suffixes, and postfixes. Thus, an Arabic word can have a more complicated form if all these affixes are attached to its root. If we could remove all affixes of a word, then we will get a stemmed word that is not the root but a basic word without any affixes and so we maintain the meaning of the word and improve the search effectiveness. In this research, we apply light stemming and Khoja stemmer on our datasets.

Term weighting is one of the pre-processing methods used for enhanced text document representation. It helps to locate important terms in a document collection for ranking purposes [40]. There are several term weighting schemes: Boolean model, Term Frequency (TF), Inverse Document Frequency (IDF), and Term Frequency-Inverse Document Frequency (TF-IDF) [12]. Choosing an appropriate term weighting scheme is important for text categorization [41]. Term Frequency and Inverse Document Frequency (TF-IDF) is a popular method of preprocessing documents in the information retrieval community [41]. In this research, TF-IDF term weighting is applied to our datasets.

The basic measures that we use to evaluate our models are accuracy, precision, recall, and F-measure.

IV. Experimental Results

The combined models are implemented using two data mining tools, Weka [42] and RapidMiner [43]. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from Java code. Weka contains tools for data pre-processing, classification, regression, clustering,

association rules, and visualization. It is also well suited for developing new machine learning schemes. RapidMiner provides data mining and machine learning procedures including data loading and transformation, data preprocessing and visualization, modeling, evaluation, and deployment. RapidMiner is written in Java. It uses learning schemes and attributes evaluators from the Weka machine-learning environment. We use Weka tool to build two models using fixed combining rules and stacking. The AdaBoost and bagging models are built using RapidMiner. In our implementation, we use a platform of Intel Core i3 of speed 2.2 GHz, 4 GB of memory and 64-bit Windows 7 operating system.

A. Arabic Text Documents Classification using Fixed Combining Rules

In this model, we use the following five fixed combination rules: average rule, product rule, majority voting, minimum rule, and maximum rule. We apply these rules using different number of classification algorithms; we combine three, five and seven classifiers. Almost, we use three datasets to confirm our results. We use TF-IDF term weighting.

In the first step, we use three classifiers with each rule; the classifiers used are SVM, Naive Bayes, and C4.5. Table 4 shows the results when applying light stemming on the BBC Arabic dataset. The table shows that the majority voting rule achieved the highest accuracy (94.1%), recall (0.943), precision (0.943), and F-measure (0.943). Table 5 shows the results when applying Khoja stemmer on BBC Arabic dataset. Also, we notice from this table that the majority voting rule outperforms all other fixed combination rules. Table 6 shows the results when applying light stemming on the CNN Arabic dataset. These results also confirm our previous results on the BBC Arabic data. Table 7 shows the results when applying Khoja stemmer on CNN Arabic dataset. From Tables 4 to 7, we notice that applying the majority voting rule on BBC Arabic dataset with light stemming gives the highest accuracy (94.1%). Also, using the average rule with BBC Arabic dataset with Khoja stemmer needs lesser time (91 s) than any other model, and this model gives an accuracy of 92.9% as shown in Table 5.

Table 4. Performance of Three Combined Classifiers and BBC Dataset, Light Stemming: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.937	0.938	0.937	93.7	91
Product	0.940	0.940	0.941	90.4	105
Majority	0.943	0.943	0.943	94.1	107
Minimum	0.895	0.893	0.897	89.6	110
Maximum	0.880	0.879	0.882	88.2	112

Table 5. Performance of Three Combined Classifiers and BBC Dataset, Khoja Stemmer: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.929	0.929	0.929	92.9	135
Product	0.911	0.912	0.911	91.3	156
Majority	0.935	0.935	0.935	93.5	154
Minimum	0.894	0.891	0.898	90.7	157
Maximum	0.861	0.860	0.862	86.2	166

Table 6. Performance of Three Combined Classifiers and CNN Dataset, Light Stemming: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.924	0.925	0.924	92.4	469
Product	0.923	0.923	0.923	92.3	460
Majority	0.959	0.959	0.959	93.4	457
Minimum	0.909	0.909	0.910	91.2	452
Maximum	0.884	0.890	0.879	87.9	454

Table 7. Performance of Three Combined Classifiers and CNN Dataset, Khoja Stemmer: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.922	0.923	0.922	92.2	308
Product	0.910	0.911	0.910	91.8	292
Majority	0.924	0.925	0.924	92.4	300
Minimum	0.909	0.911	0.908	91.8	298
Maximum	0.877	0.888	0.867	86.7	298

Next, we combine five classifiers using fixed rules combination method. The classifiers used are SVM, Naive Bayes, C4.5, kNN, and Decision Stump. Table 8 shows the results when applying light stemming on the BBC Arabic dataset. The table shows that the majority voting rule achieves the highest accuracy (94.5%), recall (0.945), precision (0.945), and F-measure (0.943). From Table 9, we notice also that the majority voting rule outperforms all other fixed combination rules using Khoja stemmer. To confirm our results, we use another dataset as shown in Table 10 which shows the results when applying light stemming on the CNN Arabic dataset. Table 11 shows the results when applying Khoja stemmer on CNN Arabic dataset. From Tables 8 to 11, we notice that the best results are obtained when using majority voting rule applied on BBC Arabic dataset with light stemming. Also, we notice that building average rule model using BBC Arabic dataset with Khoja stemmer needs lesser time (90 s) than any other model, and this model gives an accuracy of 92.7% as shown in Table 9.

Table 8. Performance of Five Combined Classifiers and BBC Dataset, Light Stemming: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.925	0.926	0.925	92.5	151
Product	0.911	0.912	0.911	91.3	167
Majority	0.945	0.945	0.945	94.5	159
Minimum	0.915	0.913	0.917	91.4	148
Maximum	0.861	0.860	0.862	86.2	166

Table 9. Performance of Five Combined Classifiers and BBC Dataset, Khoja Stemmer: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.928	0.929	0.927	92.7	90
Product	0.912	0.913	0.912	91.1	91
Majority	0.943	0.944	0.943	94.3	96
Minimum	0.942	0.943	0.942	89.6	91
Maximum	0.880	0.879	0.882	88.2	94

Table 10. Performance of Five Combined Classifiers and CNN Dataset, Light Stemming: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.930	0.930	0.930	92.9	489
Product	0.926	0.926	0.926	92.5	478
Majority	0.954	0.959	0.950	93.4	473
Minimum	0.905	0.909	0.901	90.4	463
Maximum	0.884	0.879	0.890	87.9	509

Table 11. Performance of Five Combined Classifiers and CNN Dataset, Khoja Stemmer: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.921	0.922	0.921	92.1	299
Product	0.950	0.950	0.950	91.8	301
Majority	0.926	0.926	0.926	92.6	299
Minimum	0.902	0.903	0.902	90.7	300
Maximum	0.877	0.888	0.867	86.7	281

Now, we combine seven classifiers using fixed rules combination method; the classifiers used are SVM, Naive Bayes, C4.5, RBFN, kNN, Decision Stump, and Nearest-neighbor-like. Table 12 shows the results when applying light stemming on the BBC Arabic dataset. We notice that the accuracy increases when applying majority voting rule on BBC Arabic dataset; the highest accuracy is 95.3%. Also, we apply the model on BBC Arabic dataset with Khoja stemmer. Table 13 shows that the maximum accuracy is obtained using the majority voting rule. Table 14 shows the results when we use CNN Arabic dataset with light stemming, and we notice that the majority voting accuracy is 93.3% which is the highest one over all other rules. Table 15 shows the results when using CNN Arabic dataset with Khoja stemmer, which confirm all previous results in which the majority voting gives the highest accuracy. From Tables 12 to 15, we notice that the best results are obtained using the majority voting rule applied on BBC Arabic dataset with light stemming. Also, we notice that building majority voting model using BBC Arabic dataset with Khoja stemmer needs lesser time (323 s) than any other model, and this model gives a high accuracy (92.8%) as shown in Table 15.

From the previous tables, we can conclude that the best accuracy is achieved when using a seven-classifiers model using BBC dataset with light stemming.

Table 12. Performance of Seven Combined Classifiers and BBC Dataset, Light Stemming: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.935	0.939	0.932	93.2	730
Product	0.926	0.926	0.926	92.9	788
Majority	0.954	0.955	0.953	95.3	836
Minimum	0.901	0.901	0.901	90.2	735
Maximum	0.876	0.883	0.870	87.0	741

Table 13. Performance of Seven Combined Classifiers and BBC Dataset, Khoja Stemmer: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.920	0.920	0.921	92.0	361
Product	0.910	0.908	0.912	90.1	361
Majority	0.946	0.946	0.946	94.6	324
Minimum	0.901	0.901	0.902	89.2	361
Maximum	0.864	0.870	0.858	85.8	332

Table 14. Performance of Seven Combined Classifiers and CNN Dataset, Light Stemming: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.921	0. 922	0.920	92.0	5949
Product	0.917	0.916	0.918	91.8	6102
Majority	0.930	0.931	0.930	93.3	5946
Minimum	0.901	0.902	0.901	90.9	5892
Maximum	0.890	0.889	0.891	88.9	5982

Table 15. Performance of Seven Combined Classifiers and CNN dataset, Khoja Stemmer: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Rule	F	P	R	A	T
Average	0.919	0.919	0.919	91.9	3682
Product	0.901	0.902	0.901	90.0	3723
Majority	0.926	0.927	0.926	92.8	3603
Minimum	0.887	0.887	0.887	88.9	3590
Maximum	0.861	0.861	0.861	86.0	3584

B. Arabic Text Documents Classification using Stacking

In the stacking approach, we use two basic models, where the difference between them is the meta classifier. In the first model, we use Naïve Bayes classifier as a meta classifier, while in the second model we use linear

regression prediction as a meta classifier. In each model, we use a different number of base classifiers. The base classifiers that we use in all models are Naïve Bayes, SVM, C4.5, Decision Stump, k-Nearest Neighbor (kNN), radial basis function network (RBFN), and Learning Vector Quantization (LVQ). We implement the models using three and five base classifiers. We could not implement the stacking model with seven classifiers since it needs higher resources. The models are evaluated using two datasets: BBC and CNN, with two different stemming algorithms (light Stemming and Khoja stemmer).

First, we build combined models that are based on Naïve Bayes classifier as a meta classifier using three and five base classifiers. The first model that we evaluate is a stacking model that consists of the following three base classifiers: Naïve Bayes, SVM, and C4.5. Three different datasets are used to confirm the results as shown in Table 16. The highest accuracy is obtained using BBC Arabic dataset with light stemming (98.9 %). Table 17 shows the results of combining LVQ, Naive Bayes, and C4.5 using stacking with Naïve Bayes as a meta classifier. The results show that stacking these classifiers gives a high classification accuracy with two datasets.

The third model that we evaluate is a stacking model that consists of the following five base classifiers: SVM, Naive Bayes, C4.5, Decision Stump, and kNN. From Table 18, we notice that we get a very high accuracy using five classifiers, but also the time needed to build the model is increasing compared to that of Table 16. We use only two datasets because using stacking with five base classifiers needs high memory recourses, so we could not use OSAC dataset with stacked models that contain five classifiers.

The fourth model that we evaluate is a stacking model that consists of the following five base classifiers: LVQ, Naive Bayes, C4.5, RBF networks, and kNN. Table 19 shows the results using Naïve Bayes as meta classifier. Also, we notice that the accuracy increases when using five base classifiers compared to the model that contains only three classifiers.

Table 16. Performance of Stacked Model of Three Classifiers (Naïve Bayes, SVM and C4.5) and Naïve Bayes Meta Classifier: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.989	0.989	0.989	98.9	1480
BBC with Khoja Stemmer	0.975	0.976	0.975	97.6	951
CNN with Light stemming	0.924	0.924	0.924	92.4	3672
CNN with Khoja Stemmer	0.906	0.907	0.906	90.8	3204
OSAC with Light stemming	0.968	0.968	0.968	96.8	16434
OSAC with Khoja Stemmer	0.956	0.956	0.957	95.7	15527

Table 17. Performance of Stacked Model of Three Classifiers (LVQ, Naive Bayes and C4.5) and Naïve Bayes Meta Classifier: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light stemming	0.982	0.983	0.982	98.3	1714
BBC with Khoja Stemmer	0.977	0.979	0.976	97.7	1536
CNN with Light Stemming	0.960	0.962	0.959	96.0	3504
CNN with Khoja Stemmer	0.951	0.952	0.950	95.1	3254

Table 18. Performance of Stacked Model of Five Classifiers (SVM, Naive Bayes, C4.5, Decision Stump and kNN) and Naïve Bayes Meta Classifier: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.992	0.993	0.992	99.2	1963
BBC with Khoja Stemmer	0.983	0.985	0.981	98.9	1761
CNN with Light Stemming	0.977	0.978	0.977	97.8	3757
CNN with Khoja Stemmer	0.965	0.966	0.964	96.4	3572

Table 19. Performance of Stacked Model of Five Classifiers (LVQ, Naive Bayes, C4.5, RBF Networks and kNN) and Naïve Bayes Meta Classifier: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.988	0.989	0.988	98.9	2163
BBC with Khoja Stemmer	0.983	0.985	0.981	98.1	1823
CNN with Light Stemming	0.975	0.976	0.975	97.6	4197
CNN with Khoja Stemmer	0.970	0.970	0.970	96.9	3714

Then, we built combined models that are based on linear regression as a meta classifier using three and five base classifiers. The first is a stacking model that consists of the following three base classifiers: Naïve Bayes, SVM, and C4.5. Table 20 shows the results; we notice that this model achieves a high accuracy using BBC dataset. The second is a stacking model that consists of the following five base classifiers with linear regression as a meta classifier: Naïve Bayes, SVM, C4.5, Decision Stump, and kNN. Table 21 shows the results; we notice the accuracy increases when using five classifiers and the time also increases.

C. Arabic Text Documents Classification using Boosting and Bagging

We built a model that uses AdaBoost to classify Arabic text documents. The model is based on C4.5 classifier. Table 22 shows the results with 5 iterations; we notice that we get the highest accuracy (95.3%) using BBC Arabic dataset with light stemming. From Tables 22 and 23 we see that increasing the number of iterations produces a higher classification accuracy using all datasets.

Table 20. Performance of Stacked Model of Three Classifiers (Naïve Bayes, SVM and C4.5) and Linear Regression Meta Classifier: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.994	0.994	0.994	99.4	1568
BBC with Khoja Stemmer	0.977	0.977	0.977	97.7	1026
CNN with Light Stemming	0.938	0.939	0.938	93.9	3702
CNN with Khoja Stemmer	0.922	0.922	0.922	92.2	3290
OSAC with Light stemming	0.955	0.956	0.955	95.6	16964
OSAC with Khoja Stemmer	0.942	0.942	0.942	94.2	15892

Table 21. Performance of Stacked Model of Five Classifiers (SVM, Naive Bayes, C4.5, Decision Stump and kNN) and Linear Regression Meta Classifier: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.994	0.995	0.994	99.5	3718
BBC with Khoja Stemmer	0.980	0.980	0.980	98.0	3291
CNN with Light stemming	0.940	0.941	0.940	94.0	8722
CNN with Khoja Stemmer	0.929	0.929	0.929	93.0	8037

Table 22. Performance of using AdaBoost with C4.5 Classifier using 5 Iterations: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.952	0.953	0.952	95.3	1175
BBC with Khoja Stemmer	0.941	0.941	0.942	94.2	922
CNN with Light Stemming	0.924	0.924	0.924	92.6	3544
CNN with Khoja Stemmer	0.901	0.901	0.901	90.1	3329

Table 23. Performance of using AdaBoost with C4.5 Classifier using 10 Iterations: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.995	0.995	0.995	99.5	1966
BBC with Khoja Stemmer	0.980	0.980	0.980	98.0	1585
CNN with Light stemming	0.942	0.942	0.943	94.3	4878
CNN with Khoja Stemmer	0.938	0.938	0.939	93.9	4398

We built a model that uses bagging to classify Arabic text documents. The model is based on Decision Tree classifier with 5 iterations as shown in Table 24. We notice that the highest accuracy is obtained when using BBC dataset with light stemming. We repeated the same experiment but with 10 iterations as shown in Table 25. We notice that we get a higher accuracy when increasing the number of iterations.

Table 24. Performance of using Bagging with Decision Tree using 5 Iterations: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.936	0.936	0.936	93.7	296
BBC with Khoja Stemmer	0.930	0.931	0.930	93.0	201
CNN with Light Stemming	0.911	0.912	0.911	91.1	922
CNN with Khoja Stemmer	0.906	0.906	0.906	90.6	739

Table 25. Performance of using Bagging with Decision Tree using 10 Iterations: F = F-Measure, P = Precision, R = Recall, A = Accuracy (%), T = Time (s)

Dataset	F	P	R	A	T
BBC with Light Stemming	0.993	0.993	0.993	99.3	471
BBC with Khoja Stemmer	0.977	0.977	0.977	97.7	366
CNN with light Stemming	0.928	0.929	0.928	92.9	1428
CNN with Khoja Stemmer	0.913	0.913	0.913	91.3	1132

D. Comparing Combined Models with Single Classifiers

Fig. 1 compares the accuracy of combining three classifiers using majority voting rule and BBC Arabic dataset with light stemming with other single classifiers. We notice that the combined model of three classifiers (Naïve Bayes, SVM and C4.5) gives the highest accuracy (94.1%) [16, 18, 21].

Fig. 2 compares the accuracy of combining five classifiers using majority voting rule and BBC Arabic dataset with light stemming with other single classifiers. The combined model of five classifiers (Naïve Bayes, SVM, C4.5, kNN and Decision Stump) gives the highest accuracy (94.5%) [10, 16, 18, 21].

Fig. 3 compares the accuracy of combining seven classifiers using majority voting rule and BBC Arabic dataset with light stemming with other classifiers. The comparison shows that the combined model using seven classifiers (Naïve Bayes, SVM, C4.5, kNN, RBFN, Nearest-neighbor-like, and Decision Stump) yields the highest accuracy [10, 16, 18, 21].

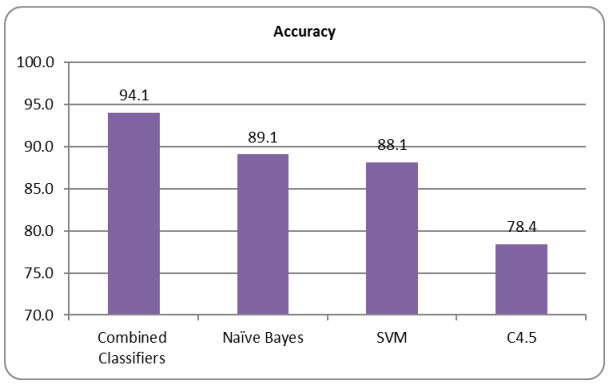

Fig.1. A comparison between three combined classifiers using majority voting rule vs. single classifiers.

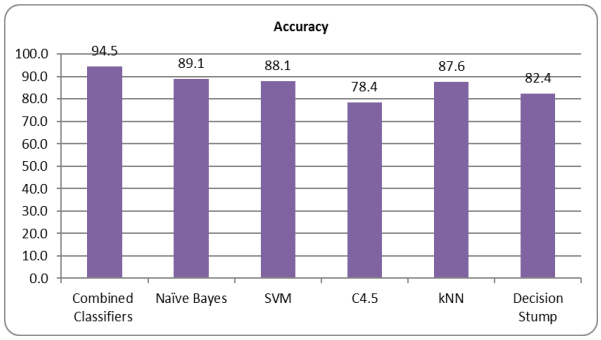

Fig.2. A comparison between five combined classifiers using majority voting rule vs. single classifiers.

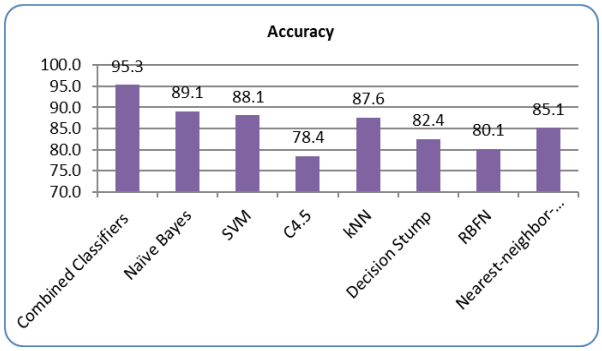

Fig.3. A comparison between seven combined classifiers using majority voting rule vs. single classifiers.

Fig. 4 compares between a stacked model using three classifiers (Naïve Bayes, SVM and C4.5) and Naïve Bayes as a meta classifier and single classifiers. We see that the accuracy of the stacked model (98.3%) is higher than any single classifier. In Fig. 5, we use three other classifiers (LVQ, Naïve Bayes and C4.5); the comparison shows that the stacked model outperforms other single classifiers [16, 18, 44].

The other comparison is done between stacked models built using five classifiers. The first model consists of Naïve Bayes, kNN, SVM, Decision Stump, and C4.5 using Naïve Bayes as a meta classifier. Fig. 6 shows that the accuracy of the model exceeds that of any single classifier. Fig. 7 shows another model that consists of Naïve Bayes, kNN, LVQ, Decision Stump, and C4.5. Also, stacking outperforms all single classifiers used in [10, 16, 18, 21].

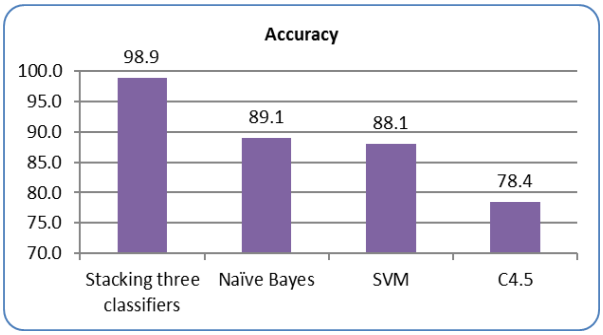

Fig.4. A comparison between stacking using three classifiers vs. single classifiers.

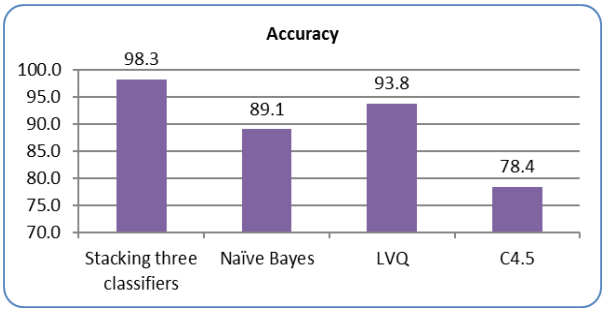

Fig.5. A comparison between stacking using three classifiers vs. single classifiers.

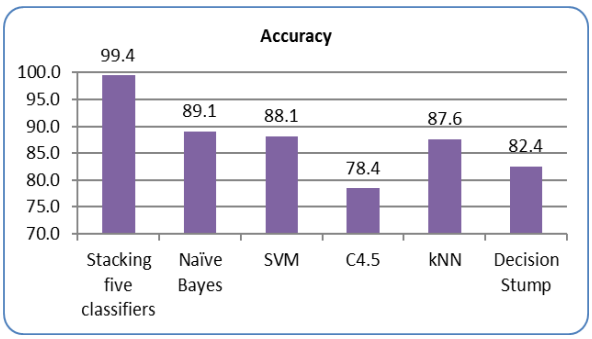

Fig.6. A comparison between stacking using five classifiers vs. single classifiers.

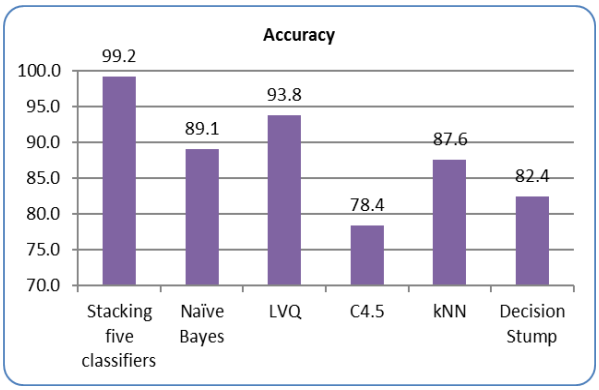

Fig.7. A comparison between stacking using five classifiers vs. single classifiers.

Fig. 8 shows the comparison between stacked models using linear regression as a meta classifier and other single classifiers. The results show that using five stacked classifiers gives a higher accuracy compared to single classifiers used in [10, 16, 18, 21, 44].

From Fig. 9, we see that using C4.5 as a single classifier used in [10] achieved a classification accuracy of 78.42%, which is low compared to using the same classifier with boosting which improves the accuracy to 99.5% using 10 iterations.

We use Decision Tree classifier with bagging; first we implement bagging using 5 and 10 iterations. Fig. 10 shows the results. We notice that the accuracy is improved compared to that using the Decision Tree as a single classifier such as in [45].

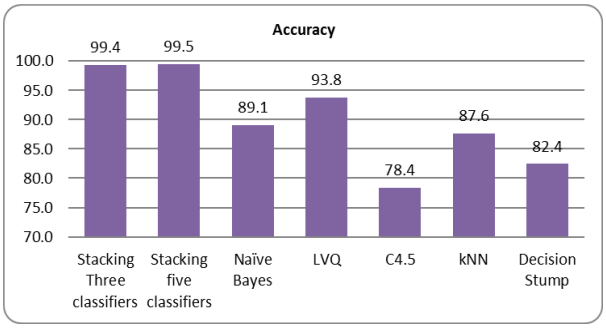

Fig.8. A comparison between stacking using three and five classifiers vs. single classifiers.

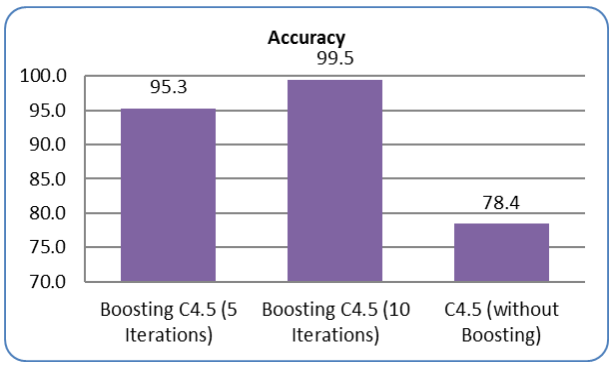

Fig.9. A comparison between AdaBoost vs. single classifier (C4.5) using 5 and 10 iterations.

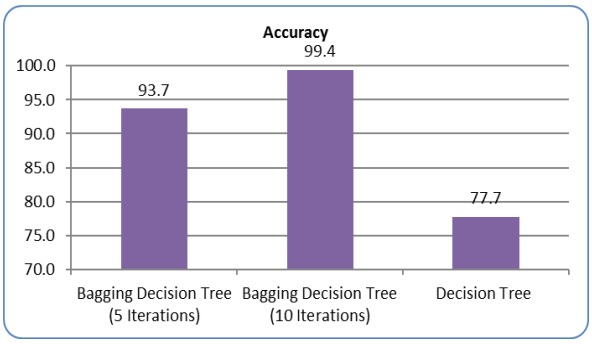

Fig.10. A comparison between bagging vs. single classifier (Decision Tree) using 5 and 10 iterations.

E. Discussion

In this paper, we implement different approaches to combine classifiers. Fixed combining rules, stacking, boosting, and bagging are used to improve the accuracy of classifying Arabic text document. Two stemmers are applied on our datasets, where we use only TF-IDF term weighting.

All combined approaches used in this paper achieve a high accuracy. The best results are obtained using BBC Arabic dataset with light stemming. The time needed to build the models depends on the combination algorithm; we notice that stacking needs more time than any other approach. Also, as the number of classifiers used in the model increases the time needed to build the model also increases.

The fixed combination rules need less time to build the model than any other combination approach, because this approach simply calls a non-trainable combiner. Each classifier in the model gives a decision and the combiner uses the selected rule to give the final decision.

The stacking algorithm needs more time than any other combination approach because it consists of two levels of classifiers. The first level consists of base classifiers that are trained using the dataset, and then the outputs of the base classifiers are used with the original dataset to produce a new dataset. 70 % of the dataset is used to train the base classifiers, and 30% for testing. The second level or meta classifier is trained using the new dataset by using 10 folds cross validation. The 10 folds cross validation training method needs more time than the method used with the base classifiers. Due to all of this, the stacking approach needs more time to be built.

The AdaBoost algorithm also achieves a very high accuracy in classification. The AdaBoost algorithm focuses on the unclassified or misclassified documents during building the model. It assigns weights and focuses on these documents through the next iterations to improve the classification accuracy. The AdaBoost algorithm requires an acceptable time to build the model compared with the stacking model.

The last approach is bagging which achieves a high accuracy. We notice that bagging needs lesser time than AdaBoost algorithm to build the model because bagging trains different models at the same time and combines their decisions.

Table 26 shows a comparison between fixed combination rules and stacking when using the BBC Arabic dataset. We notice that the accuracy of using three classifiers with fixed combining rules is 94.1%, and the needed time to build the model is 107 seconds. When we increased the number of classifiers to seven, the accuracy increased to 95.3%, but on the other side, the time needed to build the model increased to 836 seconds. Therefore, increasing the accuracy by 1.2% needs additional 729 seconds.

Table 26. Comparing the Accuracy and Time between Fixed Combining Rules and Stacking

Number of Classifiers	Fixed Combining Rules		Stacking	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
3 classifiers	94.1	107	98.9	1480
5 classifiers	94.5	159	99.2	1963
7 classifiers	95.3	836	-	-

The accuracy of the stacking algorithm is 98.9%, and the time needed to build the model is 1480 seconds when using three classifiers. However, when we used five classifiers, the accuracy is 99.2% and the time needed to build the model is 1964 seconds. Comparing the results of stacking model using three classifiers with the model that was built by the three classifiers using fixed combining rules, we notice that the accuracy of the stacking algorithm is higher by 4.8%, but the time cost is very high because the stacking algorithm needs 1373 seconds more than the fixed combining rules combiner. This is because stacking uses two levels of learning and 10 fold cross validation learning method to train the meta

classifier, while the fixed combining rules do not need to train the combiner.

Table 27 shows a comparison between AdaBoost and bagging algorithms. We notice that AdaBoost needs more time than bagging due to its iterative nature to build the model. For example, using ten iterations, the accuracy of AdaBoost is 99.5% and it needs 1175 seconds to classify documents, while the accuracy of bagging algorithm is 99.3% and it needs 471 second to classify documents.

Based on the previous experimental results, we have demonstrated that combining classifiers using different approaches can effectively improve the accuracy of classifying Arabic Text documents.

Table 27. Comparing the Accuracy and Time between AdaBoost and Bagging

Number of Iterations	AdaBoost		Bagging	
	Accuracy (%)	Time(s)	Accuracy (%)	Time(s)
5	95.3	1175	93.7	296
10	99.5	1966	99.3	471

V. Conclusion

In this paper, classifiers are combined into four models to classify Arabic text documents. The first model used fixed combining rules. The second one is stacking which used two stages of classification, where in the first stage it used base classifiers, and in the second one it trains a meta classifier based on the results of base classifiers to give the final classification result. The third and the fourth models are AdaBoost and bagging, respectively, where we used different number of iterations in each one.

In our experiments, we used three datasets: BBC Arabic, CCN Arabic, and OSAC datasets. We used two stemming methods: light stemming and Khoja stemmer, with TF-IDF term weighting.

The results of combining classifiers using fixed combining rules showed that the majority voting rule outperformed all single classifiers that were used to build the model and it outperformed all other fixed combining rules such as average of probability, median of probability, and other fixed rules. The highest accuracy was achieved by majority voting using BBC Arabic dataset with light stemming. The accuracy of the model using seven classifiers is 95.3% which is high compared to using a single classifier. The time required to build this model is 836 seconds which was relatively acceptable compared to some single classifiers such as Decision Tree. We used different classifiers for this model. The results showed that the accuracy increased when we increase the number of classifiers, but at the same time, the required time to build the model increases also. The accuracy using a model with three classifiers is 94.1%, and it is 94.5% when using five classifiers, but the best accuracy, 95.3%, is achieved using seven classifiers.

The second model that we built is a stacking model, where the accuracy was very high compared to that of single classifiers; but it requires more time to build the model because stacking needs two stages to train the model. The first stage is to train the base classifiers; and the second stage is to train the meta classifier based on the original dataset and the classification results of the base classifiers. We used Naïve Bayes and linear regression as meta classifiers to build our stacked model, and we used three and five base classifiers for each model. The best results are achieved using Naïve Bayes meta classifier when using five base classifiers, 99.2% accuracy; while the best accuracy when using linear regression with five base classifiers is 99.4%. These results are achieved using BBC Arabic dataset with light stemming.

The third model is built using AdaBoost with C4.5 classifier. The AdaBoost improved the performance of C4.5 classifier. Boosting the C4.5 using 5 iterations achieved 95.3% accuracy, and 99.5% accuracy when using 10 iterations.

The last model we built is by using bagging with Decision Tree. The model achieved a high accuracy and improved the results of decision tree classifier. The results showed that using 5 iterations achieved an accuracy of 93.7%, and 99.4% accuracy using 10 iterations.

In all previous models, the highest achieved accuracy is using BBC Arabic dataset with light stemming. The combined models were compared to other single classifiers that are used by researchers to classify Arabic text documents. The comparison was done in terms of accuracy, precision, recall, and F-measure. The combined models that we built in our research improved the accuracy of classifying Arabic text documents. All models achieved a high classification accuracy compared to the single classifiers used by other researchers; although some models such as stacking needed more time to be built.

There are limitations in our experiments. The first limitation is that we cannot use the OSAC dataset with all models because the OSAC dataset did not fit into memory especially with stacking, AdaBoost, and bagging. The second limitation is that we cannot build a stacking model that consists of more than five classifiers because of required high memory resources, and at the same time increasing the number of base classifiers produced a model that needed a very long time to be built.

Fixed combining rules, AdaBoost, and bagging achieved a high accuracy and needed an acceptable time to build the model compared to some classification algorithms.

According to the results of experiments and the limitations that we faced, efforts can be devoted in the future to investigate the following points:

- Using classifiers other than those used in this research to build a combined model by fixed rules to achieve better results.
- Reducing the time needed to build a combined model especially for stacked models.
- Adopting our models to deal with large datasets specially when using a large number of classifiers.

ACKNOWLEDGMENT

We thank anonymous referees for their constructive comments.

REFERENCES

[1] T. David and D. Robert, "Experiments with Classifier Combining Rules," in Proceedings of the First International Workshop on Multiple Classifier Systems, Cagliari, Italy, 2000.

[2] T. Dietterich, "Ensemble Methods in Machine Learning," in Proceedings of the First International Workshop on Multiple Classifier Systems, London, UK, 2000.

[3] G. Fumera and F. Roli, "A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp. 942–956, 2005.

[4] M. Ponti, "Combining Classifiers: From the Creation of Ensembles to the Decision Fusion," in 4th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials, Sao Carlos, Brazil, 2011.

[5] R. Lior, Pattern Classification Using Ensemble Methods, New Jersey: World Scientific Publishing Co. Pte. Ltd., 2010.

[6] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On Combining Classifiers," IEEE Trans Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226–239, 1998.

[7] D. Wolpert, "Stacked Generalization," Neural Networks, vol. 5, no. 2, pp. 241-259, 1992.

[8] J. Quinlan, "Bagging, Boosting, and C4.5," in In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 1996.

[9] B. Leo, "Bagging Predictors," Machine Learning, vol. 24, no. 2, pp. 123-140, 1996.

[10] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. Khorsheed, and A. Al-Rajeh, "Automatic Arabic Text Classification," in Proceedings of The 9th International Conference on the Statistical Analysis of Textual Data, Lyon-, France, 2008.

[11] R. Duwairi, "Arabic Text Categorization," The International Arab Journal of Information Technology, vol. 4, no. 2, pp. 125-132, 2007.

[12] M. Saad and W. Ashour, "Arabic Text Classification Using Decision Trees," in Computer science and information technologies, Moscow, Russia, 2010.

[13] M. Saad, "The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification," Master Thesis, The Islamic University, Gaza, 2010.

[14] A. El-Halees, "A Comparative Study on Arabic Text Classification," Egyptian Computer Science Journal, vol. 20, no. 2, 2008.

[15] G. Kanaan, R. Al-Shalabi, S. Ghwanmeh, and H. Al-Ma'adeed, "A Comparison of Text Classification Techniques Applied to Arabic Text," Journal of the American Society for Information Science and Technology, vol. 60, no. 9, pp. 1836–1844, 2009.

[16] A. Mesleh, "Chi Square Feature Extraction Based SVMs Arabic Language Text Categorization System," Journal of Computer Science, vol. 3, no. 6, pp. 430-435, 2007.

[17] F. Harrag and E. El-Qawasmeh, "Neural Network for Arabic Text Classification," in the second International Conference of Applications of Digital Information and Web Technologies, London, 2009.

[18] M. El-Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," in the 20th International Conference on Computational Linguistics, Geneva, 2004.

[19] A. El-Halees, "Arabic Text Classification Using Maximum Entropy," The Islamic University Journal, vol. 15, no. 1, pp. 157-167, 2007.

[20] H. Sawaf, J. Zaplo, and H. Ney, "Statistical Classification Methods for Arabic News Articles," in In the Workshop on Arabic Natural Language Processing, Toulouse, France, 2001.

[21] R. Al-Shalabi, G. Kanaan, and M. Gharaibeh, "Arabic Text Categorization using kNN Algorithm," in Proceedings of the 4th International Multi-conference on Computer Science and Information Technology, Amman, Jordan, 2006.

[22] I. Hmeidi, B. Hawashin, and E. El-Qawasmeh, "Performance of KNN and SVM Classifiers on Full Word Arabic Articles," Advanced Engineering Informatics, vol. 22, no. 1, pp. 106–111, 2008.

[23] M. Abbas, K. Smaili, and D. Berkani, "Comparing TR-Classifier and kNN by using Reduced Sizes of Vocabularies," in The 3rd International Conference on Arabic Language Processing, Rabat, Morroco, 2009.

[24] M. Bawaneh, M. Alkoffash, and A. Al-Rabea, "Arabic Text Classification using K-NN and Naive Bayes," Journal of Computer Science, vol. 4, no. 7, pp. 600-605, 2008.

[25] A. El-Halees, "Arabic Opinion Mining Using Combined Classification Approach," in Proceedings of the International Arab Conference on Information Technology, Azrqa, Jordan, 2011.

[26] A. Danesh, B. Moshiri, and O. Fatemi, "Improved Text Classification Accuracy Based on Classifier Fusion Methods," in Proceedings of The 10th International Conference on Information Fusion, Quebec, Canada, 2007.

[27] A. Fujino, H. Isozaki, and J. Suzuki, "Multi-label Text Categorization with Model Combination based on F1-score Maximization," in Proceedings of the 3rd International Joint Conference on Natural Language Processing, Kyoto, Japan, 2008.

[28] Y. Bi, D. Bell, H. Wang, G. Guo, and J. Juan, "Combining Multiple Classifiers Using Dempster's Rule of Combination for Text Categorization," Applied Artificial Intelligence, vol. 21, no. 3, pp. 211-239, 2007.

[29] M. Saad, "Arabic Computational Linguistics," 26 07 2010. [Online]. Available: http://sourceforge.net/projects/ar-text-mining/. [Accessed 23 04 2013].

[30] A. Fahad, A. Ibrahim, and F. Salah, "Processing Large Arabic Text Corpora: Preliminary Analysis and Results," in Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt, 2009.

[31] M. Attia, "Arabic Tokenization System," in Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Prague, Czech Republic, 2007.

[32] T. Gharib, M. Habib, and Z. Fayed, "Arabic Text Classification Using Support Vector Machines," International Journal of Computers and Their Applications, vol. 16, no. 4, pp. 192-199, 2009.

[33] M. R. Al-Maimani, A. Naamany, and A. Z. A. Bakar, "Arabic Information Retrieval: Techniques, Tools and Challenges," in GCC Conference and Exhibition, 2011.

[34] Hayder K. Al Ameed, Shaikha O. Al Ketbi, Amna A. Al Kaabi, Khadija S. Al Shebli, Naila F. Al Shamsi, Noura H. Al Nuaimi, and Shaikha S. Al Muhairi, "Arabic Light Stemmer: A new Enhanced Approach," in The Second International Conference on Innovations in Information

Technology (IIT'05), Dubai, 2005.
[35] C. Aitao, "Building an Arabic Stemmer for Information Retrieval," in Proceedings of the Eleventh Text Retrieval Conference, Berkeley, 2003.
[36] S. Khoja and R. Garside, "Stemming Arabic Text," in Lancaster, UK, Computing Department, Lancaster University, 1999.
[37] M. Ababneh, R. Al-Shalabi, G. Kanaan, and A. Al-Nobani, "Building an Effective Rule-Based Light Stemmer for Arabic Language to Improve Search Effectiveness," The International Arab Journal of Information Technology, vol. 9, no. 4, pp. 368-372, 2012.
[38] N. Abdusalam, S. Tahaghoghi, and S. Falk, "Stemming Arabic Conjunctions and Prepositions," in Proceedings of the 12th international conference on String Processing and Information Retrieval, Heidelberg, 2005.
[39] L. Leah, B. Lisa and C. Margaret, "Light Stemming for Arabic Information Retrieval," Arabic Computational Morphology Text, Speech and Language Technology, vol. 38, pp. 221-243, 2007.
[40] Q. Zhengwei, G. Cathal, D. Aiden, and S. Alan, "Term Weighting Approaches for Mining Significant Locations from Personal Location Logs," in CIT 2010 - 10th IEEE International Conference on Computer and Information Technology, Bradford, UK, 2010.
[41] L. Man, T. Chew-Lim, L. Hwee-Boon, and S. Sam-Yuan, "A Comprehensive Comparative Study on Term Weighting Schemes for Text Categorization with Support Vector Machines," in WWW '05 Special interest tracks and posters of the 14th international conference on World Wide Web, Chiba, Japan, 2005.
[42] "Weka 3: Data Mining Software in Java," Machine Learning Group at the University of Waikato, [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/.
[43] "RapidMiner," [Online]. Available: http://rapid-i.com/.
[44] M. Azara, T. Fatayer, and A. El-Halees, "Arabic Text Classification using Learning Vector Quantization," in 8th International Conference on Informatics and Systems (INFOS), Giza, Egypt, 2012.
[45] F. Harrag, E. El-Qawasmeh and P. Pichappan, "Improving Arabic Text Categorization using Decision Trees," in First International Conference on Networked Digital Technologies, Ostrava, 2009.

Authors' Profiles

Ibrahim S. I. Abuhaiba is a professor at the Islamic University of Gaza, Computer Engineering Department. He obtained his Master of Philosophy and Doctorate of Philosophy from Britain in the field of document understanding and pattern recognition. His research interests include artificial intelligence, computer vision, image processing, document analysis and understanding, pattern recognition, information security, and computer networks. Prof. Abuhaiba published tens of original contributions in these fields in well-reputed international journals and conferences.

Hassan M. Dawoud received his B.Sc. degree in computer engineering, Islamic University of Gaza in 2003, and master degree in computer engineering, Islamic University of Gaza, in 2014. He research interests include artificial intelligence.

How to cite this paper: Ibrahim S. I. Abuhaiba, Hassan M. Dawoud,"Combining Different Approaches to Improve Arabic Text Documents Classification", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.4, pp.39-52, 2017. DOI: 10.5815/ijisa.2017.04.05