Arabic text classification using master-slaves technique

Zinah Abdulridha Abutiheen¹, Ahmed H. Aliwy² and Kadhim B. S. Aljanabi²

¹Department of Computer Science, Faculty of Science, University of Kerbala, Kerbala, Iraq
²Department of Computer Science, Faculty of CS and Mathematics, University of Kufa, Kufa, Iraq

z.aboaltaheen@uokerbala.edu.iq

Abstract. Text classification (TC) is an essential field in both text mining (TM) and natural language processing (NLP). Humans have a tendency to organize and categorize everything as they want to make things easier to understand. Therefore, text classification is an important step to achieve this goal. Arabic text classification (ATC) is a difficult process because the Arabic language has complications and limitations resulting from the nature of its morphology. In this paper, a proposed approach called the Master-Slaves technique (MST) was used to improve Arabic text classification. It consists of two main phases: in the first phase, a new Arabic corpus of 16757 text files was collected. These text files were classified into five categories manually. In the second phase, four different classifiers were implemented on the collected corpus. These classifiers are Naïve Bayes (NB), K-Nearest Neighbour (KNN), Multinomial Logistic Regression (MLR) and Maximum Weight (MW). Naïve Bayes classifier was implemented as Master and the others as Slaves. The results of these slave classifiers were used to change the probability of the Naïve Bayes classifier (Master). The four classifiers used were implemented individually and the simple voting technique was implemented among them too on the collected corpus to check the effectiveness and efficiency of the proposed technique. All the tests were applied after the pre-processing of Arabic text documents (tokenization, stemming, and stop-word removal) and each document was represented as vector of weights. For the reliability of the results, 10-fold cross-validation was used in this paper. The results showed that the Master-slaves technique gives a good improvement in accuracy of text document classification with accepted algorithm complexity compared to other techniques.

Key words: Arabic corpus, text classification, text pre-processing, Arabic text classification, combined classifier.

1. Introduction

Text Classification (TC) is a very useful task in many applications of natural language processing (NLP) and text mining (TM) like information retrieval (IR). It is used as a basis for employment in text processing, web mining, text filtering (as email spam), text organization (as documents archive), and so on. TC (also known as text categorization, topic spotting, document categorization or document classification) is a process of assigning and labeling documents to a set of predefined categories based on their contents [1]. The categories may be literature, arts, economy, sport, etc.

The nature of Arabic language morphology is very complicated. Thus, classification is very difficult and requires time because the language needs preprocessing[2]. The preprocessing is done twice, the first time for data set and the second time for preparing it to input to classifiers. Many
classifiers used for Arabic documents classification such as Decision Trees \[3, 4\], K-Nearest Neighbor\[5\], Naïve Bayes\[6\], Maximum Entropy \[7, 8\] and others which give good accuracy.

2. Related Work

Almost all the research carried out in this field showed that the results obtained from the combination of classifiers are much better than the results obtained by the same classifiers individually for text categorization.

In \[9\] proposed the combination of three classifiers: K-Nearest Neighbor (KNN), Relevance feedback and Bayesian independence classifiers on the corpus that consists of 11,599 documents divided into a training set of 10,902 documents and a testing set of 187 documents in the medical domain for automatic assignment of ICD9 codes. The performance of the classifiers was measured on the basis of document ranks. The result showed improving accuracy.

In \[10\] combined many monolingual classifiers (the sum of SVMs output values, F1 weighted sum of SVMs output values, F1 weighted sum of SVMs decisions) on multilingual text classification. A set of 2714 documents, two Anglo-Saxon languages (English and German) and two Roman (Italian and Portuguese) were used. Each document was represented by using the bag-of-words approach, a vector space model (VSM) and TF-IDF weighting function. The result of combined classifiers showed better accuracy than the single classifier.

In \[11\] combined classification approach: first, lexicon based method is used which classifies some documents, then the classified documents are used as training set for maximum entropy model which subsequently classified some other documents. After that, k-nearest model was used to classify the rest of the documents. The documents collected in Arabic from three different domains: "education", "politics" and "sports" forum, and a total of 1143 posts contain 8793 Arabic statements with average of 7.7 statements in each post were used. The accuracy was moved (almost) from 50% when using only lexicon based method to 60% when using lexicon based method and maximum entropy together, to 80% when using the three combined methods.

In \[12\] combined probabilistic classifiers (Naïve Bayes classifier and Maximum Entropy). They used two merging operators, max and harmonic mean. The proposed method was evaluated using the "ModApte" split of the Reuters-21578 data set. The evaluation results showed a measurable improvement in the final evaluation accuracy.

In \[13\] combined Naïve Bayes and modified Maximum Entropy classifiers. Reuters-21578 and the DB-World data set were used. During this work, the proposed combination classifier with max combining operator gives the best accuracy.

In \[14\] used a hybrid classifier to analyze text review. Naïve Bayes and MaxEnt classifiers have been combined using combining operations like mean, max, and average etc. These classifiers were applied to the Blitzer’s unprocessed data set and showed a better accuracy.

In \[15\] built four models by combining different approaches to improve Arabic text documents classification. The first combined model is built using fixed combination rules. The best classification accuracy, 95.3%, is achieved by using majority voting rule with seven classifiers. The second combination approach is stacking. In their experiments, they used different numbers of base classifiers and two different meta classifiers: Naïve Bayes and linear regression. Stacking achieved very high classification accuracy, 99.2% and 99.4%, using Naïve Bayes and linear regression as meta classifiers, respectively. The third model they used 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, while the accuracy is 99.5% using 10 iterations. The fourth model uses bagging with decision tree. The accuracy is 93.7% achieved when using 5 iterations, and 99.4% when using 10 iterations. They used three datasets, to test the combined models, BBC Arabic of 4763 text files, CNN Arabic of 5070 text files, and OSAC datasets of 22429 text files.

This work, compared to the related works, focuses on: (i) introducing the technique called the Master-Slaves technique for solving the problems of classifying Arabic text documents (ii)
constructing a new Arabic corpus taken from Iraqi media, which can be used by researchers in the future, (iii) testing the corpus with four classifiers: NB, KNN, LR and MW, (iv) comparing the results of six classifiers.

3. Pre-processing
Preprocessing represents the first step in TC in which the required documents are prepared as input to the classification process by applying machine learning techniques with high efficiency and accuracy. Documents should, firstly, be preprocessed which is an important stage in TC.

The aim of pre-processing is to reduce dimensions and to distinguish between the most important documents by selecting the significant or relative words and ignore the irrelevant words. This will turn the document into more suitable format.

![Algorithm: Pre-processing](image)

After cleaning and organizing the corpus files, they must be preprocessed to get a new version of the documents suitable for the next phase. In other words, the corpus must be suitable to input for machine learning. The general idea of preprocessing is explained in algorithm below.

Where in Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. In Stop-Words Removal is simply removing the words that occur commonly across all the documents in the corpus and unimportant words that appear in a text. In Stemming is the process for reducing inflected words to their word stem (base form) which can be removed any affixes (prefixes that added to the beginning of the word, infixes that added to the middle of the word, and suffixes that added to the ending of the word) from the words to reduce these words to their stems or roots under the assumption that words sharing the same stem. In Weight is calculated from term frequency (TF) and inverse document frequency (IDF) as follows:

\[ \text{Weight} (t) = tf \times idf \quad \ldots (1) \]

Where:
\( tf \): Term Frequency of term \( t \) in document \( d \) is defined as the number of times that \( t \) occurs in \( d \).
\( idf \): Inverse Document Frequency estimates the rarity of a term in the whole document collection. (If a term occurs in all the documents of the collection, its IDF is zero.)

\[ \text{Idf} = \log \left( \frac{\text{Number of documents}}{\text{Number of documents that contain word } t} \right) \quad \ldots (2) \]

Each word with idf equal to zero is deleted when computing weights, thus reduces the size of the text file.

4. Classifiers
Many classification techniques can be used to classify texts such as Naïve Bayes, K-Nearest Neighbor and many others. Like other tasks of NLP, TC needs preprocessing (tokenization, stop-word removal, normalization and stemming). Supervised machine learning techniques used in this research include labeling texts for training and predefining texts for testing to evaluate the performance of the classifiers. In this work, four classifiers were used, which are Naïve Bayes, K-Nearest Neighbor, Logistic Regression, and Maximum Weights.

4.1. Naïve Bayes

Naïve Bayes (NB) classifier is a simple probabilistic classifier based on Bayes’ theorem with strong independence assumptions between the features. For this reason, it is called naive. NB is the most commonly used generative classifier[16]. In general, most researchers employ the NB method by applying Bayes rule:

\[
C_{\text{MAP}} = \arg\max_{c_j \in C} \hat{p}(c_j) \prod_i \hat{p}(x_i | c_j) \tag{3}
\]

Where:

\[
\hat{p}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{\text{doc}}} \tag{4}
\]

\[
\hat{p}(x_i | c_j) = \frac{\text{count}(x_i, c_j)}{\sum_{x \in V} \text{count}(x, c_j)} \tag{5}
\]

Where

\(p(c_j)\): probability of class \(c_j\) among set of classes \(C\),
\(p(x_i|c_j)\): probability that the term \(x_i\) occurs in class \(c_j\) which may be zero in training data, so the smoothing technique (as Laplace) is chosen to estimate it[17].

There are two classes of models commonly used for NB classification based on the distribution of the words in the document. The major difference between these classes is an assumption in terms of taking (or not taking) word frequencies into account, in the Multinomial Model (MNB), the document is represented by a bag of words and frequencies of terms are computed in it[18]. The probability of occurrence of each feature \(i\) in a category \(c\) can computed as \(p(x_i|y)\)[19]. Another class is Multivariate Bernoulli Model (BNB), in which the presence or absence of words in a text document can be used as features to represent a document. Thus, the word features in the text are assumed to be binary[18].

The Naïve Bayes is used as a master with modifying probabilities of the classes or the features for a given class in the Master-Slaves technique.

4.2. K-Nearest Neighbor

The k-nearest neighbor algorithm (k-NN) is a statistical learning algorithm based on the distance or similarity function for pairs of observations, such as the Euclidean distance or Cosine similarity measures between “k” training data and the testing data. The degree of similarity between documents and k training data depend on a value of k [20]. The best choice of k depends upon the data. Usually, the greater value of k reduces the effect of noise on the classification, but it makes the boundaries between classes less distinct. If k is greater than one, then the object is classified by a majority voting of its neighbors according to the value of k, if \(k = 1\), then the object is assigned to the class of its nearest neighbour [21].

4.3. Multinomial Logistic Regression

Multinomial Logistic Regression (MLR) is also known as Log-linear Models (linear if log is taken), and a conditional exponential classifier or logistic regression classifier. It includes polytomous LR,
multiclass LR, softmax regression, multinomial logit, maximum entropy (MaxEnt) classifier and conditional maximum entropy model [5, 18].

Multinomial logistic regression modelling is a general and an intuitive way for estimating a probability from the data and it has been applied successfully in various natural language processing tasks [22].

Conditional distributed \( p(c|d) \) is computed as follows:

\[
p(c|d) = \frac{1}{Z} \exp \left( \sum_i w_i f_i(c, d) \right)
\]

(6)

Where \( Z(d) \) is the normalization function that is computed as:

\[
Z(d) = \sum c \exp(\sum_i \alpha f_i(d, c))
\]

(7)

\( \alpha \) is a weight of term; \( f_i \) is a feature.

4.4. Maximum Weights

Maximum Weight (MW) is a new classifier suggested by Zena Abd Al-Retha Abo-Alaheet. It is a very simple method for text classification, which works by selecting the highest weight of the term among the categories and only these values are used to predict the best class for any input example. Formally:

Let classes \( C = \{c_1, c_2, \ldots, c_n\} \)

Suppose that:

\( W_{\text{imax}}: \) is the maximum weight of the term \( i \) among all these classes.

\( C_{ij}: \) is a binary value which indicates the class \( j \) and contains the maximum weight of the term \( i \)

\[
c_{ij} = \begin{cases} 1, & \text{if } W_{\text{imax}} \in c_j \\
0, & \text{otherwise} \end{cases}
\]

(8)

Then the best class can be estimated by:

\[
c_{\text{best}} = \arg \max_{c_j \in C} \sum_{i=1}^{m} c_{ij} * W_{\text{imax}}
\]

(9)

Where:

\( m = \) set of words in the test document.

Obviously, this method depends on features; each feature in the testing data is searched in all categories. Maximum weight in the training data in one class is taken, and then the maximum frequency of the category is computed as a result.

This method can be implemented easily, where each feature is associated with one category only. Each class has a subset of all features set in the dataset, so the feature reduction does not matter in this method.

4.5. Voting Technique

This classifier works on building multiple models (typically of different types), and simple statistics (like calculating the mean) are used to combine predictions. This work depends on NB, MW, MLR, and KNN in its results.

4.6. Master-Slaves Technique

Master-Slaves technique (MST) is suggested in [23],[24] on parts of speech. In this paper, this classifier was updated and implemented on Arabic text classification. MST consists of one classifier as a master and several other classifiers as slaves. NB is selected as the master and MLR, KNN and MW as the slaves. The master classifier modifies its probability according to the results of slaves by multiplying each probability by a factor according to Eq. 8. This factor reflects the weight of those slaves.

The master classifier multiplies the probability of all features with category except the results of KNN or MLR or MW. In other words, it takes the result of KNN classifier, multiplies the probability of each feature by a factor except this result and repeats this multiplication in accumulative for other
slaves classifiers. Thus, it gives a result that depends on three classifiers and modifies the probability of document with each class.

\[ f_i = 1 - A_i + C \] (10)

Where \( f_i \) is the factor for slave \( i \), \( A_i \) is the accuracy of slave \( i \), and \( c \) is a constant value less than 0.5 used because the eq. (10) may give a very small number; may be less than 0.2.

The Master-Slaves technique has been explained in Figure 1 below:

![Figure 1. Master-Slaves Technique](image)

5. Data Set
The corpus is built from “Al-Sabah” newspaper. 16757 Arabic documents are used for the first time in supervised machine learning. This corpus is organized manually into five different categories as in figure 2. Dataset is divided into two parts: 90% as training data and 10% as testing data.

6. Implementation and Results
Initially, two text classification algorithms (KNN, and NB) are applied on the new Arabic corpus by two scenarios:
- The first scenario is conducted by merging all documents under the same category into one file, and then the classifier is applied to evaluate the relationship between test document and the entire category.
- The second scenario is achieved by taking each document separately and the similarity is calculated with each file. Then the average of all documents similarity is calculated to classify the test document and see to which class it belongs.

The results calculated by the average are 86.495% and 84.30% for first and second scenarios respectively. The higher results are obtained from the first scenario so it will be relied on in the next tests.

All six classifiers are implemented in a 10-fold cross-validation. The results are explained in table 1:
- Each classifier was applied separately and tested alone.
- The four classifiers are combined in the voting technique.
- These classifiers are used in master-slaves technique where NB represents the master and the other three classifiers as slaves with constant factor \( f=0.1 \). This test called is Master_Slaves1.
- A second test called Master-Slaves2 was conducted using the variable factor \( f_i \), according to Eq. 10.
e. The experiments showed that MST can be used for document classification. It is also in control of the result of classification even if found very weak as a result of a classifier. That is, there is no dropping at all in the accuracy results from pure classifier.

f. MST always makes the classification with a new probability and the training process is new. This probability is always different and therefore the training is not fixed, and this procedure does not exist in any other way in the mixed classifier.

Table 1. Results of Implementing Six Different Classifiers.

| #Fold | # training files | # testing files | NB  | KNN  | LR   | MW   | Voting | Master Slaves1 | Master Slaves2 |
|-------|-----------------|----------------|-----|------|------|------|--------|----------------|----------------|
| 1     | 15080           | 1677           | 85.45 | 78.903 | 76.937 | 81.585 | 82.41 | 86.524 | 86.524 |
| 2     | 15080           | 1677           | 89.326 | 81.63 | 77.34 | 83.13 | 83.96 | 89.45 | 89.326 |
| 3     | 15080           | 1677           | 87.12 | 79.73 | 76.63 | 82.11 | 82.41 | 87.359 | 87.24 |
| 4     | 15080           | 1677           | 78.295 | 72.81 | 69.53 | 74.84 | 75.194 | 79.43 | 79.43 |
| 5     | 15080           | 1677           | 89.92 | 83.07 | 79.96 | 85.39 | 85.927 | 90.46 | 90.28 |
| 6     | 15082           | 1675           | 90.042 | 81.515 | 76.923 | 83.363 | 84.079 | 90.34 | 90.28 |
| 7     | 15082           | 1675           | 91.711 | 84.258 | 77.877 | 85.868 | 86.524 | 91.95 | 91.77 |
| 8     | 15082           | 1675           | 89.267 | 80.739 | 78.354 | 83.304 | 83.781 | 89.922 | 89.624 |
| 9     | 15082           | 1675           | 88.67 | 82.409 | 80.143 | 84.496 | 85.092 | 89.863 | 89.863 |
| 10    | 15082           | 1675           | 88.193 | 80.382 | 77.818 | 82.469 | 82.887 | 88.611 | 88.611 |
| Average |                 |                | 87.80 | 80.54 | 77.15 | 82.66 | 83.23 | 88.39 | 88.29 |

7. Conclusion
The method that used in Arabic text classification (Master-Slaves Technique) for first time can be used for document classification as showed in the results. It also improves the performance of documents classification. As we can see, the accuracy of Master-Slave technique is much better than the other classification techniques (single and combined classifiers).

We can see that there is no dropping at all in the accuracy results of the Master classifier when compared to pure classifier results. It still has the control for selecting the right class.

Finally, Master Slaves always makes the classification with a new probability and the training process is new. This probability is always different and therefore the training is not fixed, and this procedure does not exist in any other way in the mixed classifier.

8. References
A. Abu-Errub, "Arabic Text Classification Algorithm using TFIDF and Chi Square Measurements," *International Journal of Computer Applications*, vol. 93, 2014.

A. Al-Badarenah, E. Al-Shawakfa, K. Al-Rababah, S. Shatnawi, and B. Bani-Ismail, "Classifying Arabic text using KNN classifier," 2016.

S. Bahassine, M. Kissi, and A. Madani, "New stemming for Arabic text classification using feature selection and decision trees," in Proceedings of the 5th International Conference on Arabic Language Processing, 2014, pp. 26-27.

H. F. Witschel, "Using decision trees and text mining techniques for extending taxonomies," in Learning and Extending Lexical Ontologies by using Machine Learning Methods, Workshop at ICML-05, 2005, pp. 61-68.

R. Alhutaish and N. Omar, "Arabic text classification using k-nearest neighbour algorithm," Int. Arab J. Inf. Technol.(IAJIT), vol. 12, pp. 190-195, 2015.

N. Sharma, "CLASSIFICATION USING NAÏVE BAYES-A SURVEY."

A. M. El-Halees, "Arabic text classification using maximum entropy," *IUG Journal of Natural Studies*, vol. 15, 2015.

S. Zhu, X. Ji, W. Xu, and Y. Gong, "Multi-labelled classification using maximum entropy method," in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005, pp. 274-281.

L. S. Larkey and W. B. Croft, "Combining classifiers in text categorization," in Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, 1996, pp. 289-297.

T. Gonalves and P. Quresma, "Multilingual text classification through combination of monolingual classifiers," in Proceedings of IV Workshop on Legal Ontologies and Artificial Intelligence Techniques, 2010, p. 29.

A. El-Halees, "Arabic opinion mining using combined classification approach," 2011.

K. Fragos, P. Belsis, and C. Skourlas, "Combining Probabilistic Classifiers for Text Classification," Procedia-Social and Behavioral Sciences, vol. 147, pp. 307-312, 2014.

H. Kashyap and B. Buksh, "Combining Naïve Bayes and Modified Maximum Entropy Classifiers for Text Classification," 2016.

M. Shah, S. Shinde, R. S. Sawant, and P. Wagh, "Analysis of Text Review using Hybrid Classifier," *International Journal of Engineering Science*, vol. 10914, 2017.

I. S. Abuhaiba and H. M. Dawoud, "Combining Different Approaches to Improve Arabic Text Documents Classification," *International Journal of Intelligent Systems and Applications*, vol. 9, p. 39, 2017.

a. K. K. Nisha Mariam Daniel, "Survey on Text Classification Methods," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, pp. 585--588, 2016.

D. Shen, Z. Chen, Q. Yang, H.-J. Zeng, B. Zhang, Y. Lu, et al., "Web-page classification through summarization," in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 242-249.

C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," *Mining text data*, pp. 163-222, 2012.

M. Usman, Z. Shafique, S. Ayub, and K. Malik, "Urdu Text Classification using Majority Voting," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 7, pp. 265-273, 2016.

A. Khan, B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of advances in information technology*, vol. 1, pp. 4-20, 2010.

M. K. Saad, "The impact of text preprocessing and term weighting on arabic text classification," Gaza: Computer Engineering, the Islamic University, 2010.

A. Jain and J. Mandowara, "Text Classification by Combining Text Classifiers to Improve the
Efficiency of Classification, "International Journal of Computer Application" (2250-1797), vol. 6, 2016.

[23] A. H. Aliwy, "Arabic morphosyntactic raw text part of speech tagging system," 2013.

[24] A. H. Aliwy, "Combining POS taggers in master-slaves technique for highly inflected languages as Arabic," in Cognitive Computing and Information Processing (CCIP), 2015 International Conference on, 2015, pp. 1-5.