On the Impact of Dataset Characteristics on Arabic Document Classification

Diab Abuaiadah
Centre for Business, Information Technology & Enterprise
Waikato Institute of Technology, New Zealand

Jihad El-Sana
Computer Science Department
Ben Gurion University, Israel

Walid Abusalah
Faculty of Information Technology
Birzeit University, Palestine

ABSTRACT
This paper describes the impact of dataset characteristics on the results of Arabic document classification algorithms using TF-IDF representations. The experiments compared different stemmers, different categories and different training set sizes, and found that different dataset characteristics produced widely differing results, in one case attaining a remarkable 99% recall (accuracy). The use of a standard dataset would eliminate this variability and enable researchers to gain comparable knowledge from the published results.

General Terms
Document classification; Information retrieval; Machine learning algorithms; Stemmers

Keywords
Dataset; TF-IDF representation; Arabic Stemmers; Arabic document classification

1. INTRODUCTION
With the explosive growth of documentation on the web, information retrieval plays a crucial role for many users and vendors dealing with large datasets. Document classification – one dimension of information retrieval – involves the assignment of an electronic document to one or more predefined categories. Document classification has many applications such as document indexing, data mining, document filtering and organization.

Several information retrieval algorithms use term frequency-inverse document frequency (TF-IDF) representation, where the order of terms in the text is ignored and the frequency of terms is used as an input [1], [2]. Classification is achieved using a supervised machine learning algorithm [3] where documents are divided into two sets: a training set and a testing set. The training set is used to gather statistics and to profile each category. For each file in the testing set, the algorithm examines the content of the testing file and assigns it to the category having the maximum similarity.

Many supervised machine learning algorithms were developed to classify documents, including TF-IDF weighting scheme, Naïve Bayes, Support Vector Machine (SVM), KNN, Decision Tree and N-gram [3], [2], [4]. These algorithms were initially developed for the English language and have been adapted to other languages including Arabic.

In recent years there has been rapid growth in Arabic documentation across the web. Unlike English, little research has been undertaken regarding Arabic information retrieval [5]. Arabic is a morphologically rich and a highly inflected language; consequently many algorithms which were developed for use with English perform poorly when applied to Arabic [5].

Stemming is the process of reducing words and removing characters to enhance information retrieval [6]. Stemmers could affect the results of document classification algorithms. Salton [3] mentioned that stemming is not beneficial to classification. For Arabic; a few published papers show slightly different results for different stemmers [7]. Additionally, Al-Shammari and Lin [8] show that stemmers could improve the accuracy of automatic Arabic text processing. Evaluating the quality of Arabic stemmers using benchmarking is discussed in [9]. Arabic slang is another challenge for Arabic information retrieval as the well-known stemmers have been developed for classical Arabic. Shatnawi et al. [10] introduced a framework for stemming Arabic slang to improve search engine queries. Section 4.3 describes the results of using different stemmers and further investigates the impact of removing stop words on classification.

Another characteristic of the dataset is the number and type of categories which affect the accuracy of classification algorithms. Section 4.4, Table 3 shows that while TF-IDF implementation with a group of five categories achieved 99% accuracy, a different group of five categories but with the same implementation of TF-IDF achieved only 93.7% accuracy.

For the experiments in this research, an in-house dataset was created. It is composed of nine categories, each of which contains 300 documents. There are several versions of the dataset each corresponding to popular stemmers. All versions of the dataset are available for download from http://diab.edublogs.org/dataset-for-arabic-document-classification/. Section 3.1 provides more details regarding this dataset.

In this research the impact of dataset characteristics on classification is examined. The experiments provide results for different training sets, different stemmers and different categories. The paper is organized as follows: Section 2 presents a summary of related work; Section 3 introduces the details of the dataset and the implementation; Section 4 summarises the results of the experiments and provides an analysis; Section 5 provides conclusion and future work.

2. RELATED WORK
English language attracts the most interest for researchers in document classification throughout the world and where the Reuters dataset is used as the standard dataset [11]. Other languages, such as Arabic, receive much less attention. As there is no publicly available comprehensive dataset for Arabic document classification, individual researchers use their own in-house datasets to test the performance of several

different algorithms. Consequently the published results are based on different datasets and comparing the performances of the different algorithms is complex. Nevertheless, researchers have investigated different algorithms for Arabic document classification, which briefly presented next.

El Kourdi et al. [12] used a Naïve Bayes algorithm to classify Arabic documents and reported 68.78% accuracy (recall). The Naïve Bayes algorithm was applied to the dataset after the terms (words) were stemmed to their roots using Al-Shalabi's algorithm [13]. Mesleh [14] studied the effectiveness of six commonly used feature selection approaches and used Support Vector Machines (SVM) for classification. He did not use any stemming and claimed that empirical evidence proved that stemming added no benefit to Arabic document classification. Al-Saleem [15] showed that the Associative Classification algorithm outperformed the Support Vector Machine and the Naïve Bayes algorithms. The average accuracy for Associative Classification was 80.7%, while the accuracy for SVM and Naïve Bayes were 77.8% and 74% respectively. Khreisat [16] used N-gram frequency to illustrate that Dice measure outperforms Manhattan measure. She removed stop words, punctuation and diacritics; in some categories, the accuracy was below 50% for both Dice and Manhattan measures. El-Halees [17] used natural language techniques to pre-process the input dataset, and then applied maximum entropy. He reported 74.48% retrieval accuracy. Zahran and Kanaan [18] used Particle Swarm Optimization, a model that simulates the social behavior of bird flocks [19] for feature selection. His experiments showed that this algorithm outperforms the document frequency TF-IDF and the Chi-Squared statistical algorithm. Syiam et al. [7] examined several feature selection approaches and claim that TF-IDF is the best weighting scheme. They also reported that a hybrid of document frequency and information gain is the preferable criterion for feature selection. The proposed model showed 98% accuracy. Zaki et al. [20] used fuzzy entropy and taxonomy to improve the accuracy of Arabic document classification. Khorsheed and Al-Thubaity [21] investigated classification algorithms with a large and diverse dataset. Ababneh et al. [22] discussed variations of the vector space model using the KNN algorithm to classify Arabic documents. Zaki et al. [23] used a hybrid method of N-Grams-TF-IDF with radial basis indexing for classification.

The accuracies for the different classification algorithms ranged from 67% to 98%. The hypothesis is that the dominant factors in accuracy are the characteristics of the different datasets and not the algorithms, and in particular, the source of the data and the methodology of selecting the documents. As mentioned, the use of a standard dataset would eliminate these factors and enable researchers to make meaningful comparisons between the performances of the different algorithms.

3. DATASET, IMPLEMENTATION DETAILS AND TESTING METHODOLOGY

3.1 Dataset

The first step in building the dataset was to choose the number, type and source of categories. From browsing several published papers for Arabic document classification and from scanning well-known and reputable Arabic websites, nine major disciplines were adopted: Art, Literature, Religion, Politics, Law, Economy, Sport, Health, and Technology. For each discipline, documents were collected manually and arranged so the size of each document was about 2 Kilobytes. A document that appears to belong to more than one category was discarded from the dataset, as in this research each document is assigned to one category. For example, several documents could be assigned to the Politics category and equally could be assigned to the Sport category. Such documents were not included in this dataset. Table 1 lists the sources of each category.

Table 1. Details of the dataset

Category	No. of files	Size (MB)	Sources
Art	300	1.01	http://www.egypty.com/all-arts.aspx; http://www.elcinema.com/news/articles/2010/12/
Economy	300	1.01	http://news-all.com/; http://www.spa.gov.sa/;http://all4syria.info http://www.aljazeera.net/ebusiness/
Health	300	1.01	http://www.se77ah.com; http://www.aljazeera.net; http://www.6abib.com/
Law	300	1.02	http://www.eastlaws.com/News/NewsList.aspx; http://www.barasy.com/
Literature	300	0.98	http://www.almhml.com/;http://news-all.com/;http://adab.akhbarway.com
Politics	300	1.00	http://news-all.com/; http://www.spa.gov.sa/; http://all4syria.info http://www.aljazeera.net
Religion	300	1.08	http://news-all.com/; http://www.anbacom.com/
Sport	300	1.01	http://www.kooora.com/; http://www.soccerarabia.net/
Technology	300	1.06	http://news-all.com/;http://www.akhbarway.com/;http://www6.mashy.com

The second step is to determine the required number of documents needed in each category in order to obtain reliable results. Section 4.1 and 4.2 show that 300 documents for each category is sufficient and that increasing the number of documents provides no further benefit.

Different stemmers are used to create different versions of the dataset. The raw dataset includes the original documents (Version 1). Stop words, punctuation, and diacritics are distributed almost evenly in all categories and do not play a meaningful role in document classification. They are removed to generate the keyword dataset (Version 2). The keyword dataset was then put through a stemming procedure to generate the stemmed dataset. Since different stemmers tend to produce different datasets, several stemmed datasets were generated, each produced by different stemmers. The current dataset include three leading stemmers: Stemmed-light10 (Version 3), Stemmed-Chen (Version 4), and Stemmed-Khoja (Version 5) which were generated by the keyword dataset using light10, Chen, and Khoja stemmers, respectively.

Light10 stemmer, which is considered by many as the best stemmer for Arabic information retrieval [24], [5] strips off prefixes [ال ، وال، بال ، كال، فال، لل] and suffixes [ون ها ، ان ، ان، ين ، يه ، ية ، ه ، ة ، ي]. Chen and Gey [25] introduced another light stemmer by expanding the set of prefixes and the set of suffixes. Khoja's stemmer uses morphological analysis to extract the roots [26].

The implementations of removing the stop words, light10 and Chen stemmers are immediate using the Java programming language. Therefore, an in-house Java program was used to create Version 2, Version 3 and Version 4 of the dataset. Version 5 was created using the code for extracting the root which is freely available from Khoja's home page. (http://zeus.cs.pacificu.edu/shereen/research.htm).

Table 2 presents some properties of the different versions of the dataset. Since Arabic is a highly inflected language, the Khoja stemmer, which extracts the root, dramatically reduces the number of different terms and the average length of term.

Table 2: Details of the terms (words) in the dataset

Category	Average length of term					No. of terms (thousands)					No. of different terms (thousands)				
Version	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Art	4.69	5.28	4.04	3.78	3.21	102	68	68	68	68	28	20	11.6	9.2	3.7
Economy	5	5.55	4.12	3.89	3.19	98	64	64	64	61	22	14	8.4	6.9	2.8
Health	4.74	5.52	4.05	3.84	3.23	101	66	66	67	62	24	17	10	8.4	3.6
Law	4.96	5.66	4.07	3.86	3.16	97	67	67	64	64	24	18	10	8.1	3.12
Literature	4.97	5.56	*4.04*	3.83	3.20	94	64	64	64	61	30	22	12.4	10.2	3.9
Politics	4.93	5.59	4.15	3.90	3.29	97	64	64	64	61	28	20	12.1	9.8	3.6
Religion	4.7	5.21	3.94	3.70	3.13	106	69	69	69	62	33	25	14,7	11.9	3.8
Sport	4.83	5.49	4.24	3.93	3.38	99	67	67	67	64	20	15	10	8.2	4.1
Technology	4.99	5.61	4.23	3.97	3.32	102	67	67	67	64	26	18	10.4	8.3	3.6

3.2 Implementation Details

An in-house Java program is used to implement the Rocchio's algorithm [1]. The current implementation of this algorithm is typical as can be seen in several papers including [7]. The performance (running time) of the program has no impact on the final results and there are no particular hardware or software requirements. Figures 1 and 2 illustrate the main data structures and the main steps in the implementation of this algorithm.

In the initial stage of the program, the documents for the training set are selected randomly. Following the initial stage, the documents in the training set are used to build the TF-IDF representation for each category. Building the TF (term frequency) is straightforward. In this context, TF indicates the term frequency over all the documents in the training set which is equivalent to concatenating all documents into one single document. After calculating the TF, the values are normalized by the total size of all documents in the training set. This is followed by calculated the IDF for each category as it appears in Figure 1. The training phase ends by calculating the multiplication of TF and IDF (TF-IDF) for each category.

Figure 2 present the main steps during the testing phase. For each document in the testing set, we build the TF-IDF was built (similar to building and normalizing the TF-IDF for the training sets) and calculate the cosine similarity calculated between the TF-IDF of the document and the TF-IDF of each category. The algorithm assigns to the document the category with maximum similarity. The program tests the correctness of its decision by comparing the assigned category to the category previously assigned to this document by a human expert when the dataset was built.

N - Total number of documents in the training set.

Building IDF:

- *Scanning all terms in all files in the training set and set* $IDF[t]$ *to store the number of files containing the term t*
- *For each term t:* $IDF[t] = \log_{10} \frac{N}{IDF[t]}$

Assumption: stop words have been removed from the dataset.

Figure 1: Building IDF

Let $TFIDF_{training(C_i)}[t]$ *be the TF-IDF value for the term* t *in the training set of the category* C_i *and* $TFIDF_{testing}[t]$ *be the TF-IDF value for the term* t *in the testing file.*

- *For each category* C_i *and a testing file* f *, calculate the cosine similarity as:*

$$\cos(C_i, f) = \frac{\sum_t TFIDF_{training(C_i)}[t] \bullet TFIDF_{testing}[t]}{\sqrt{\sum_t TFIDF_{training(C_i)}[t]^2} \bullet \sqrt{\sum_t TFIDF_{testing}[t]^2}}.$$

- *Assign the file to the category with maximal cosine similarity.*

Figure 2: Using the cosine function to assign a category to a given file during the testing phase

4. EXPERIMENTS, RESULTS AND ANALYSIS

This experimental study explored the impact of various characteristics of a given dataset on classification. The results are presented in the form of recalls (accuracies) and precisions. Recall is the percentage of documents successfully classified. To determine precision, for each category the percentage of documents that have been correctly classified as belonging to that category is calculated. Precision enables us to understand which categories attract the misclassified documents. To produce reliable and valid results, each run is repeated five times and the average is calculated. Experiments show that several consecutive runs produce comparable results. The sections below provide further details.

4.1 Uniformity of the Dataset

This experiment studied the impact of choosing the training dataset on the performance of the retrieval accuracy, which is measured by the values of precision and recall. Version 3 of the dataset, which correspond to the light10 stemmer, is used. The size of the training set is 100 documents and the remaining 200 documents are used for testing. Figure 3 and Figure 4 show the values of precision and recall for five different consecutive runs which correspond to five different training sets. Overall the Religion category has the lowest recall, and the Law and Politics categories have the lowest precision, which means that these two categories attract the most misclassified documents.

As can be seen, the variation in the classification recall (precision) of the five runs for a category is inversely proportional to the classification recall (precision) of the category in general. For example, the difference in recall among the five runs in the Art category (whose recall is high) is small, while the difference in the Law category (whose recall is low) is relatively large. The highest standard deviation is less than 2% (the precision of the Law category) and the standard deviation of the average recall between the five runs is less than 1%, as shown in the last bar set of Figure 4. These results show that the dataset presents categories uniformly, and that increasing the number of documents in the dataset does not have any measurable impact on the retrieval performance.

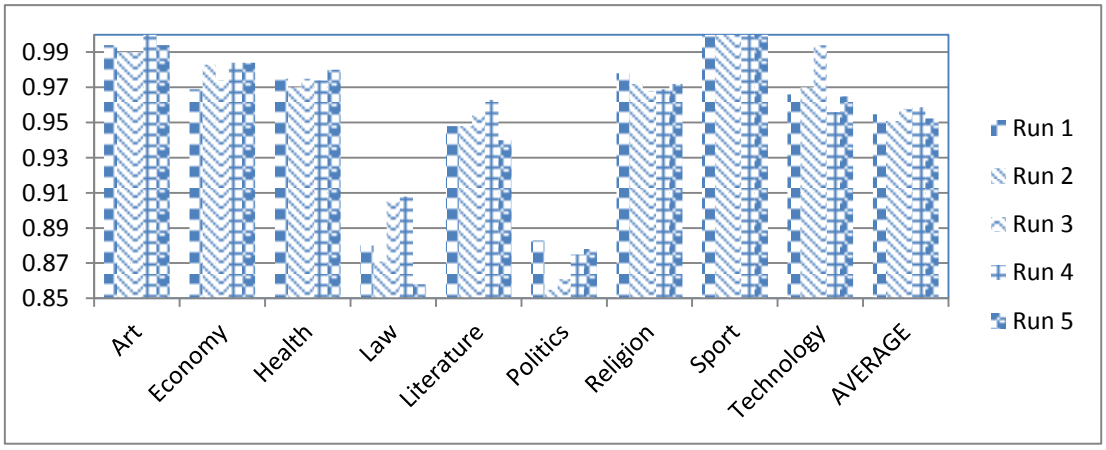

Figure 3: Precisions for five consecutive runs. Version 3 is used with 100 documents in the training set.

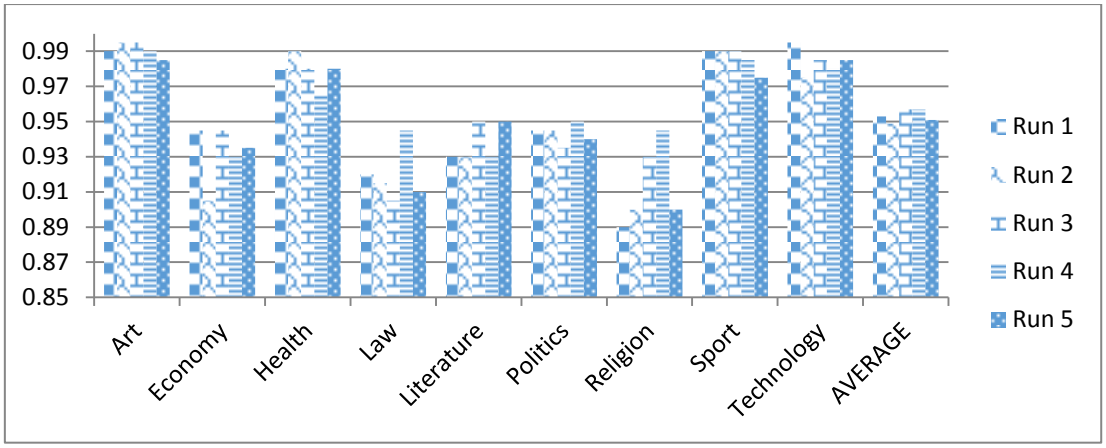

Figure 4: Accuracies for five consecutive runs. Version 3 is used with 100 documents in the training set.

4.2 Training Sets

To study the influence of training set size on the performance of the classification, the retrieval performance was tested while the size of the set was changed in a controlled manner. Figure 5 presents the values of recalls for experiments where the size of the training set began with five documents and was increased gradually to 95 documents (the remaining documents were used for the testing set). Each run was repeated five times. Similar results were achieved for the precision measure.

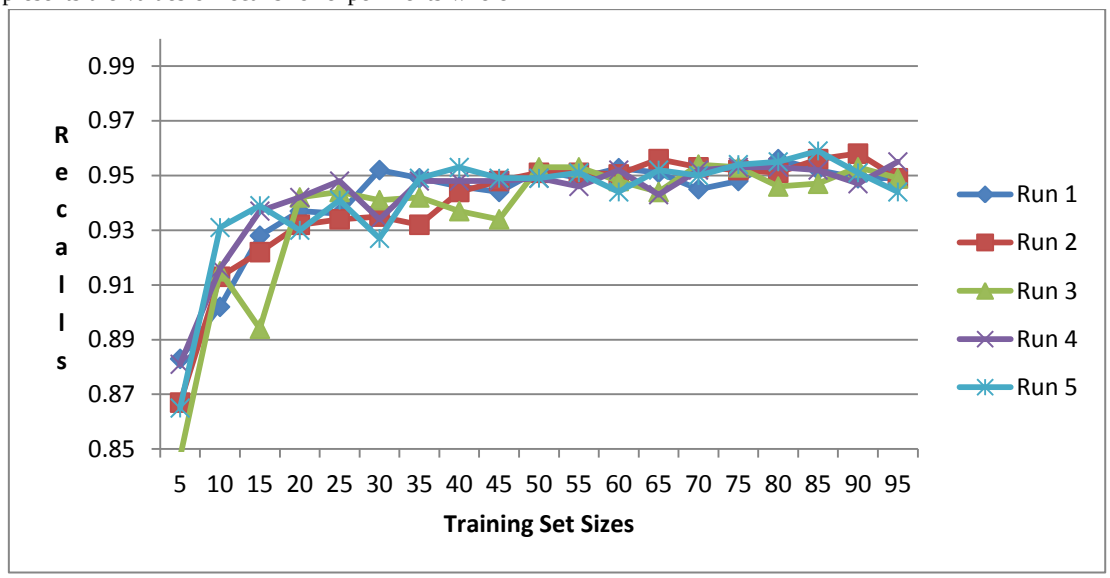

Figure 5: Recalls for different training sets with different sizes. Version 3 of the dataset is used.

As can be seen in Figure 5 the recall improves at a high rate as the size of the dataset was increased. Based on these results it is concluded that there is only a marginal improvement on the measured performance when the size of the training set exceeds 50 documents; i.e., there is no need to increase the number of documents in the dataset beyond this number to reach stable measurements. These results reiterate the previous conclusion that adding more documents to the dataset is not beneficial in this setting.

4.3 Different Stemmers and Removing Stop Words

Most applications of information retrieval remove stop words to enhance the results and decrease computational time and data storage. For the Arabic language, it is common practice to remove stop words, punctuation, and diacritics when classification algorithms are applied. However, Al-Shammari and Lin [8] suggested that the use of neglected Arabic stop words can provide a significant improvement in document processing. In addition, Al-Shargabi et al. [27] investigated the impact of stop words on different algorithms and concluded that the Support Vector Machine with sequential minimal optimization achieved the highest accuracy.

The first experiment in this Section was to evaluate the impact of removing stop words, punctuation, and diacritics. For this, the algorithm was applied to the keywords version of the dataset (Version 2) and the raw dataset (Version 1). The results are compared to the results for the raw dataset (Version 1). The size of the training set is 100 and the results are the average of five runs. Figure 6 presents the results and shows that the keywords dataset outperforms the raw data by about 2.2%, (averaged over all the categories).

Figure 7 and Figure 8 show the summary of recall and precision for the applied stemmers. There are two types of Arabic stemmers: light stemmers and root-based stemmers. Light stemmers tend to cause under-stemming errors and root-based stemmers tend to cause over-stemming errors. The Light-10 stemmer gives a small additional improvement compared to the results of the keywords dataset (Version 2); while the Chen stemmer (Version 4), which is also a light stemmer, slightly reduces the accuracy. The root-based

stemmer, Version 5, produces less accurate results compared to those of Version 2, Version 3 and Version 4. Though our observation is based on TF-IDF implementation, it is reasonable to assume that stemming might have different impacts on different classification algorithms. For example, Wahbeh et al. [28] showed that without stemming the support vector machine classifier achieved the highest classification accuracy.

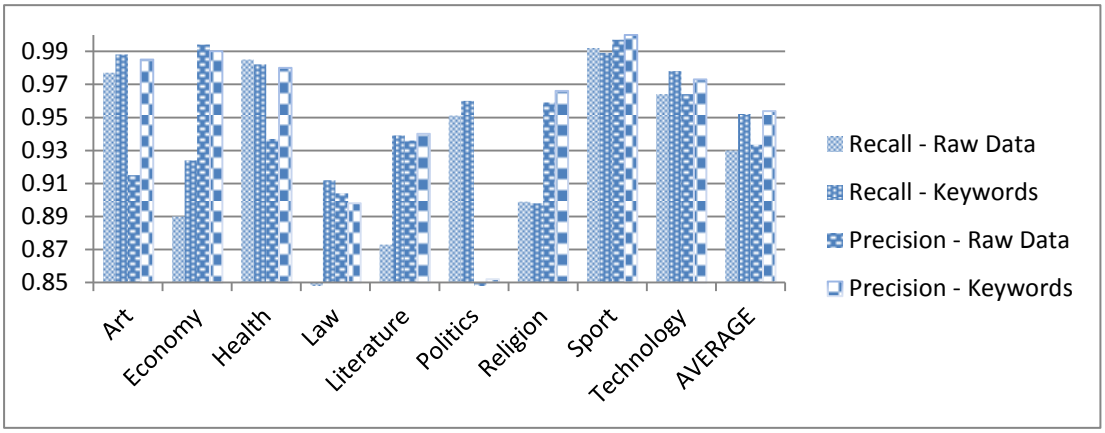

Figure 6: Accuracies and precisions for Version 1 and Version 2 of the dataset.

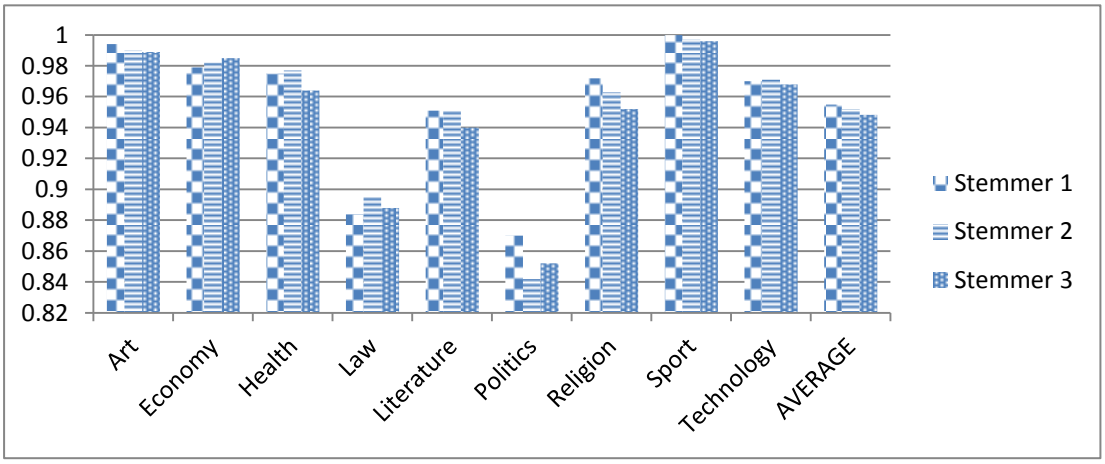

Figure 7: Precisions. Version 3 of the dataset is used with 100 documents for the training set.

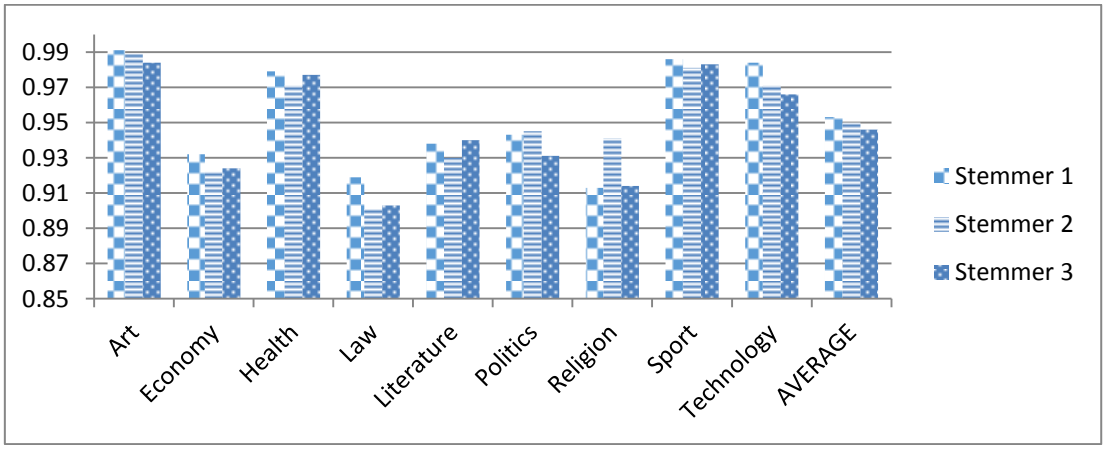

Figure 8: Accuracies. Version 3 of the dataset is used with 100 documents for the training set.

4.4 Different Categories

In this experiment the light10 stemmer (Version 3) was used with a training set that includes 100 documents. The cosine similarity function is used to measure the distances between the different categories. It is reasonable to assume that the distance between each category varies in a non-uniform manner; i.e., some are more similar to each other, while others are more different. For example, the categories Literature, Law, and Politics are similar while Sport and Religion are disparate. Categories that are more distinct are likely to produce less errors and better recalls (precisions). On the other hand, categories that are more similar are likely to produce more errors and worse recalls (precisions). Table 3 shows the recalls and precisions of different groups of categories.

Table 3: Results for different types of categories

Group A			Group B			Group C			Group D		
Category	**R**	**P**	**Category**	**R**	**P**	**Category**	**R**	**P**	**Category**	**R**	**P**
Economy	0.933	0.991	Art	1	0.999	Economy	0.885	0.885	Art	1	1
Law	0.915	0.908	Health	0.989	0.990	Law	0.935	0.894	Economy	1	0.995
Literature	0.95	0.962	Politics	0.994	0.973	Politics	0.955	0.892	Health	0.985	0.994
Politics	0.948	0.871	Sport	0.992	1	Sport	0.995	1	Religion	0.985	0.99
Religion	0.942	0.964	Technology	0.978	0.989	Technology	0.985	0.98	Sport	1	1
Average	**0.937**	**0.939**	**Average**	**0.990**	**0.990**	**Average**	**0.951**	**0.93**	**Average**	**0.996**	**0.996**

Group A contains more similar categories, and accuracies of 93.7% were achieved, while group B contains more distinct categories for which accuracies of 99% were attained. For the individual categories, the precisions do not always correlate to the recalls. For example, the category Politics (group A) has above average accuracy but has below average precision. This indicates that documents in Politics are more likely to be correctly classified compared to other categories. However, misclassified documents from other categories are more likely to be assigned to the Politics category. Another example is the Sport category in group D. The results show that all documents are correctly assigned in this category (recalls = 100%) and no misclassified document has been assigned to this category (precision = 100%).

These experiments highlight the fact that choosing the type categories, number of categories and the source of documents has a major impact on the results of classification algorithms

5. CONCLUSION AND FUTURE WORK

The experiments showed that for different dataset characteristics, the TF-IDF implementation achieved different results and, in one setup 99% accuracy was achieved. The performance of other classification algorithms might be more sensitive to dataset characteristics, although the use of a standard dataset would eliminate these factors.

For many years, the Reuters-21578 text collection system has been considered the standard dataset for the task of text categorization in the English language. Reuters-21578 is a set of 21578 documents classified according to 135 topics (categories) and is freely available for downloading.

At the present time, the field of Arabic document classification remains underdeveloped. The Arabic language has different characteristics. As mentioned above, it is desirable to build a standard dataset that would do for Arabic what Reuters-21578 has done for English. The presence of such a dataset would encourage researchers to apply several approaches that have been applied to English but have not yet been applied to Arabic.

6. REFERENCES

[1] Rocchio, J. 1971. Relevance feedback in information retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing. Englewood Cliffs, NJ, Prentice-Hall, 313-323.

[2] Salton, G. and Buckley, C. 1988. Term weighting approaches in automatic text retrieval. In Information Processing and Management, vol. 24, no. 5, 513-523.

[3] Salton, G. 1989. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Boston: Addison-Wesley Longman.

[4] Cavnar, W. and Trenkle, J. 1994. N-Gram-Based Text Categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval.

[5] Newsri, A. 2008. Effective Retrieval Techniques for Arabic Text (Doctoral dissertation). RMIT, Melbourne.

[6] Lovins, J. 1968. Development of a stemming algorithm. In Mechanical Translation and Computational Linguistics, vol. 11, 22-31.

[7] Syiam, M., Fayed, Z. and Habib, M. 2006. An Intelligent System for Arabic Text Categorization. In International Journal of Intelligent Computing and Information Sciences, vol. 6, no. 1, 1-19.

[8] Al-Shammari, E. and Lin, J. 2008. Towards an error-free Arabic stemming. In Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM-iNEWS'08).

[9] Al-Kabi, N. and Al-Radaideh, A. 2011. Benchmarking and assessing the performance of Arabic stemmers. In Journal of Information Science, vol. 37, no. 2, 111-119.

[10] Shatnawi, M., Yassein, M. and Mahafza, R. 2013. A framework for retrieving Arabic documents based on queries written in Arabic slang language. In Journal of Information Science, vol. 38, no. 4, 350-365.

[11] Lewis, D. 1997. Reuters-21578 text categorization test collection. Reuter.

[12] Elkourdi, M., Bensaid, M. and Rachidi, T. 2004. Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. In Proceedings of COLING

20th Workshop on Computational Approaches to Arabic Script-based Languages. Geneva.

[13] Al-Shalabi, R. and Evan, M. A computational morphology system for Arabic. In Proceedings of the Workshop on Computational Approaches to Semitic Languages (COLING-ACL '98), Quebec, 1998.

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

[15] Al-Saleem, M. 2010. Associative Classification to Categorize Arabic Data Sets. In The International Journal of ACM Jordan (ISSN 2078-7952), vol. 1, no. 3, 118-127.

[16] Khreisat, L. 2006. Arabic Text Classification Using N-Gram Frequency Statistics: A Comparative Study. In Proceedings of the 2006 International Conference on Data Mining, DMIN'06.

[17] El-Halees, A. 2007. Arabic Text Classification Using Maximum Entropy. In The Islamic University Journal (Series of Natural Studies and Engineering), vol. 15, no. 1, 157-167.

[18] Zahran, B. and Kanaan, G. 2009. Text Feature Selection using Particle Swarm Optimization Algorithm. In World Applied Sciences Journal, vol. 7, 69-74.

[19] Kennedy, J. and Eberhart, R. 1995. Particle Swarm Optimization. In Proc. IEEE, International Conference on Neural Networks. Piscataway.

[20] Zaki, T., Mammas, D., Ennaji, A. and Nouboud, F. 2010. Classification of Arabic Documents by a Model of Fuzzy Proximity with a Radial Basis Function. In International Journal of Future Generation, Communication and Networking, vol. 3, no. 4.

[21] Khorsheed, M. S., and Thubaity, A. O. 2013. Comparative evaluation of text classification techniques using a large diverse Arabic dataset. In Language Resources and Evaluation, vol. 47, no. 2, 513-538.

[22] Ababneh, J., Almomani, O., Hadi, W., El-Omari, N. and Al-Ibrahim, A. 2014. Vector Space Models to Classify Arabic Text. In International Journal of Computer Trends and Technology (IJCTT), vol. 7, no. 4.

[23] Zaki, T., Es-saady, Y., Mammass, D., Ennaji, A. and Nicolas, S. 2014. A Hybrid Method N-Grams-TFIDF with radial basis for indexing and classification of Arabic documents. In International Journal of Software Engineering and Its Applications, vol. 7, no. 2, 127-144.

[24] Larkey, L., Ballesteros, L. and Connell, M. 2007. Light Stemming for Arabic Information Retrieval. In Text, Speech and Language Technology, vol. 38, 221-243.

[25] Chen, A. and Gey, F. 2002. Building an Arabic stemmer for information retrieval. In NIST Special Publication 500-251: Proceedings of the Eleventh Text Retrieval Conference (TREC 2002).

[26] Khoja, S. and Garside, R. 1999. Stemming Arabic text. Lancaster University, Lancaster.

[27] Al-Shargabi, B., Olayah, F. and Al-Romimah, W. 2011. An Experimental Study for the Effect of Stop Words Elimination for Arabic Text Classification Algorithms. In International Journal of Information Technology and Web Engineering (IJITWE), vol. 6, no. 2.

[28] Wahbeh, A., Al-Kabi, M., Al-Radaidah, Q., Al-Shawakfa, E. and Alsamdi, I. 2011. The Effect of Stemming on Arabic Text Classification: An Empirical Study. In International Journal of Information Retrieval Research (IJIRR), vol. 1, no. 3.