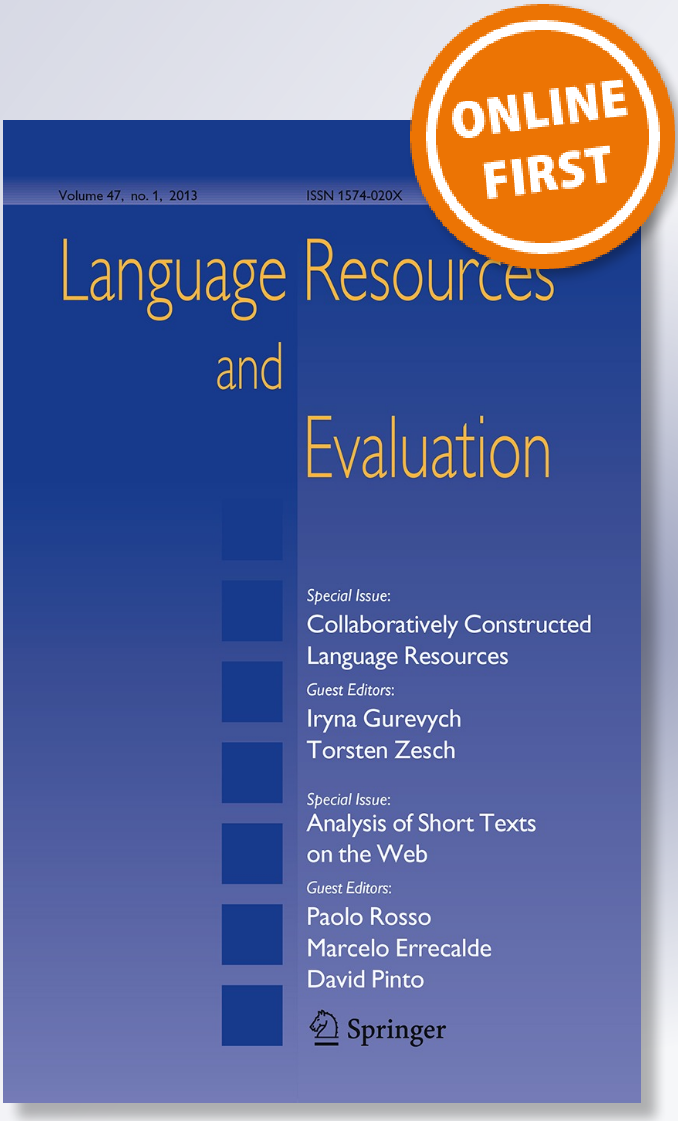
Comparative evaluation of text classification techniques using a large diverse Arabic dataset

Mohammad S. Khorsheed & Abdulmohsen O. Al-Thubaity

Language Resources and Evaluation

ISSN 1574-020X

Lang Resources & Evaluation
DOI 10.1007/s10579-013-9221-8



ONLINE FIRST

Volume 47, no. 1, 2013 ISSN 1574-020X

Language Resources and Evaluation

Special Issue:
Collaboratively Constructed Language Resources
Guest Editors:
Iryna Gurevych
Torsten Zesch

Special Issue:
Analysis of Short Texts on the Web
Guest Editors:
Paolo Rosso
Marcelo Errecalde
David Pinto

Springer

Lang Resources & Evaluation
DOI 10.1007/s10579-013-9221-8

ORIGINAL PAPER

Comparative evaluation of text classification techniques using a large diverse Arabic dataset

Mohammad S. Khorsheed · Abdulmohsen O. Al-Thubaity

Abstract A vast amount of valuable human knowledge is recorded in documents. The rapid growth in the number of machine-readable documents for public or private access necessitates the use of automatic text classification. While a lot of effort has been put into Western languages—mostly English—minimal experimentation has been done with Arabic. This paper presents, first, an up-to-date review of the work done in the field of Arabic text classification and, second, a large and diverse dataset that can be used for benchmarking Arabic text classification algorithms. The different techniques derived from the literature review are illustrated by their application to the proposed dataset. The results of various feature selections, weighting methods, and classification algorithms show, on average, the superiority of support vector machine, followed by the decision tree algorithm (C4.5) and Naïve Bayes. The best classification accuracy was 97 % for the Islamic Topics dataset, and the least accurate was 61 % for the Arabic Poems dataset.

Keywords Machine learning · Arabic text categorization ·
Arabic text classification

1 Introduction

Documents are the primary repositories of knowledge; therefore, documentation is the most effective way to illustrate ideas, thoughts, and expertise. The availability of documents in a machine-readable format and handling them in an intelligent way,

M. S. Khorsheed (✉) · A. O. Al-Thubaity
King Abdulaziz City for Science & Technology, P O Box 6086,
Riyadh 11442, Saudi Arabia
e-mail: mkhorshd@kacst.edu.sa

A. O. Al-Thubaity
e-mail: aalthubaity@kacst.edu.sa

Published online: 10 March 2013

such as through text classification, will maximize the benefit of the knowledge they contain. Arabic machine-readable texts are available both on the Internet and within government organizations and private enterprises, and they are rapidly increasing day by day. However, whereas automatic text classification is well known in natural language processing communities, little attention has been given to Arabic texts.

Text classification—the assignment of free text documents to one or more predefined categories based on their content—is used in various applications, such as e-mail filtering, spam detection, web-page content filtering, automatic message routing, automated indexing of articles, and searching for relevant information on the Web.

There are three main phases involved in building a classification system: (a) compilation of the training dataset, (b) selection of the set of features to represent the defined classes, and (c) training the chosen classification algorithm, followed by testing it using the corpus compiled in the first stage. Automated document classification involves taking a set of pre-classified documents as the training set. The training data is then analyzed in order to derive a classification scheme, which, in turn, often needs to be refined with a testing process. The derived classification scheme is then used for classification of other unknown documents. Further details will be presented in Sect. 2. The main contribution of this paper is its presentation of a large and diverse benchmarking dataset for Arabic text classification as well as an investigation of different feature selection methods, weighting methods, and text classification techniques using the same datasets.

The rest of the paper is organized as follows. Section 2 presents a brief description of text classification steps with references to some related Arabic text classification literature. In Sect. 3, the design and the statistics of the benchmarking dataset for Arabic text classification is presented in detail. The illustration of the main functions of a tool incorporated in Arabic text classification is given in Sect. 4. Sections 5, 6, 7 and 8 illustrates detailed experimentation on Arabic text classification using a set of feature selections, weighting methods, and different classifiers. Finally, discussion and some concluding remarks are presented in Sect. 9.

2 Related works

This section summarizes what has been achieved on Arabic text classification from various pieces of the literature, as shown in Table 1. The table is divided into three parts; each part is related either to data, features, or classification. Figure 1 depicts nine steps for the problem of text classification. Those steps include data collection, text processing, data division, feature extraction, feature selection, feature representation, machine learning, applying a classification model, and performance evaluation.

2.1 Data collection

Collecting data is the first step in text classification studies. The required data are samples of texts that belong to the area of interest. Each sample text must be labeled with one or more tags indicating its “belongingness” to a certain class. Some

Table 1 Techniques used in Arabic text classification [(a) data, (b) features, (c) classification]

References	1. Dataset				2. Preprocessing	3. Training/testing
	Source	Genre	# Text	# Classes		
(a)						
Sawaf et al. (2001)	1994 part of Arabic NEWSWIRE	News	N.A	34, 10	No	80/20
Elkourdi et al. (2004)	Aljazeera channel website	News	1500	5	Exclude stop words, remove diacritics	33.3/66.7, 50/50 66.7/33.3
Kanaan et al. (2005)	Two newspapers websites	News	600	6	Exclude stop words, remove digits, hyphens, and punctuation marks	25/75
Duwairi (2006)	Newspaper and magazine websites	News	1000	10	Exclude stop words, remove punctuation marks	50/50
Khreisat (2006)	Newspapers websites	News	N.A	4	Exclude stop words, remove punctuation marks, diacritics, and non-letters	40/60
Syiam et al. (2006)	Newspapers websites	News	1,132	6	Exclude stop words, remove punctuation marks, diacritics, and non-letters	N.A
Mesleh (2007)	Newspapers websites and specialised websites	News	1445	9	Exclude stop words, remove punctuation marks, diacritics, and non-letters, remove hamza	66.7/33.3
Bawaneh et al. (2008)	N.A	N.A	242	6	Exclude stop words, remove punctuation marks, diacritics, and non-letters	20-Fold cross-validation
EL-Halees (2008)	Aljazeera Channel website	News	N.A	6	Exclude stop words, remove punctuation marks, diacritics, and non-letters	k-Fold cross-validation
Thabtah et al. (2008)	Newspapers websites	News	N.A	6	No	70/30

Table 1 continued

References	1. Dataset				2. Preprocessing	3. Training/testing
	Source	Genre	# Text	# Classes		
Duwairi et al. (2009)	Different websites	N.A	15000	3	Exclude stop words, remove punctuation marks, diacritics, and non-letters	60/40
Kanaan et al. (2009)	Newspapers websites	News	1445	9	Exclude stop words, remove punctuation marks, diacritics, and non-letters	Fourfold cross-validation
Thabtah et al. (2009)	Saudi Press Agency	News	1562	6	Exclude stop words, remove punctuation marks, diacritics, and non-letters	N.A
Zahran and Kanaan (2009)	Newspapers websites	News	5183	10	Exclude stop words, remove punctuation marks, diacritics, and non-letters, normalization	60/40
Al-Saleem (2010)	Newspapers websites	News	5121	7	Exclude stop words, remove punctuation marks, diacritics, and non-letters, normalization	Tenfold cross-validation

References	4. Features type	5. Features selection	6. Features representation
(b)			
Sawaf et al. (2001)	Character n-grams	None	Relative frequency
Elkourdi et al. (2004)	Word root	TFiDF	N.A
Kanaan et al. (2005)	Word stem	Word stemming	N.A
Duwairi (2006)	Word root	Word stemming	N.A
Khreisat (2006)	Character tri-grams	No	N.A
Syiam et al. (2006)	Word stem and root	IG, OR, CHI,GSS and NGL	Boolean, TF and TFiDF

Table 1 continued

References	4. Features type	5. Features selection	6. Features representation
Mesleh (2007)	Word orthography	CHI	TFiDF
Bawaneh et al. (2008)	Word stem	Word stemming	TFiDF
EL-Halees (2008)	Word orthography	IG	N.A
Thabtah et al. (2008)	Word orthography	TFiDF, WIDF, ITF and logTF	N.A
Duwairi et al. (2009)	Word stem	Word stemming	N.A
Kanaan et al. (2009)	Word stem	Word stemming	TF, TFiDF and weighted IDF
Thabtah et al. (2009)	Word orthography	CHI	N.A
Zahran and Kanaan (2009)	Word orthography	Particle swarm optimization, CHI, DF, TFiDF	Weighted TFiDF
Al-Saleem (2010)	Word orthography	No	N.A

References	7&8. Classification algorithm	9. Evaluation
(c)		
Sawaf et al. (2001)	Maximum entropy	Precision, recall, and *f*-measure (84.2, 50.0, 62.7)
Elkourdi et al. (2004)	NB	Accuracy average (62.0 %)
Kanaan et al. (2005)	NB	Accuracy average (57.2 %)
Duwairi (2006)	Dice measure	Precision, recall, fallout, and error rate. Micro average (74.0, 62.8, 4.1, 7.4)
Khreisat (2006)	Manhattan measure and Dice measure	Precision and recall. Manhattan measure macro average (88.8, 83.1). Dice measure macro average (66.4, 56.05)
Syiam et al. (2006)	KNN and Rocchio	Precision, recall, and *f*-measure. (Results were represented in graphical form). Rocchio algorithm outperformed KNN
Mesleh (2007)	SVM	Precision, recall, and *f*-measure. Macroaverage (92.1, 84.9, 88.1)
Bawaneh et al. (2008)	NB and KNN	Accuracy. NB (73.6), KNN(84.2)
EL-Halees (2008)	Maximum entropy, NB, KNN, decision tree (DT), SVM and ANN	Precision, recall, and *f*-measure. NB outperformed all algorithms, F(83.9)

Table 1 continued

References	7&8. Classification algorithm	9. Evaluation
Thabtah et al. (2008)	KNN using cosine, dice and Jaccard	*F*-measure. macro average cosine (87.6), Dice (87.7), Jaccard (87.7). Best result for Dice TFiDF and Jaccard TFiDF
Duwairi et al. (2009)	KNN	Precision and recall. (92.0, 91.0) for light stemming
Kanaan et al. (2009)	KNN, NB and Rocchio	Precision and recall. NB outperformed others
Thabtah et al. (2009)	NB	Precision, recall, and *f*-measure (results were represented in graphical form.)
Zahran and Kanaan (2009)	Radial basis function	Precision, recall, and *f*-measure. PSO outperformed others. Macro average (90.3, 98, 93.9)
Al-Saleem (2010)	CBA, NB and SVM	Precision, recall, and *f*-measure. CBA outperformed macro average (80.5, 80.7, 80.4)

N.A Not Available

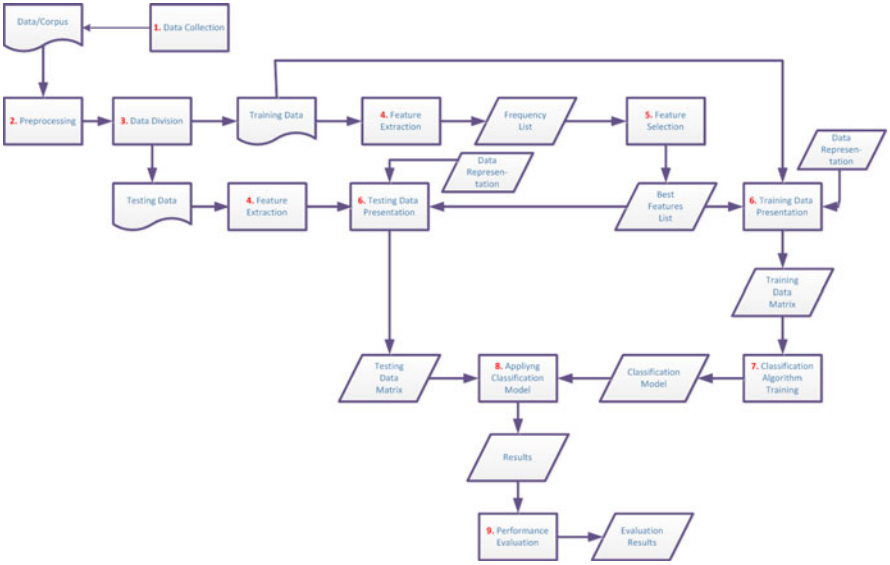


Fig. 1 Text Classification Steps

sources already label their texts, such as newspapers or press agencies. There are several free benchmarking datasets for English used for text classification, such as 20 Newsgroup, which contains around 20,000 texts distributed almost evenly into 20 classes; Reuters 21578, which contains 21,578 texts belonging to 17 classes; and RCV1 (Reuters Corpus Volume 1), which contains 806,791 texts classified into four main classes.

Unfortunately, the case is different for Arabic. There is no free benchmarking dataset for Arabic text classification. For most Arabic text classification research, authors collect their own datasets, mostly from online news sites. The collected datasets for Arabic text classification research range from 242 texts divided into six classes (Bawaneh et al. 2008) to 15,000 texts divided into three classes (Bawaneh et al. 2008). The only exception was for Sawaf et al. (2001) who used the 1994 part of Arabic NEWSWIRE. A question may arise here: what about the classification of other Arabic text genres available on the Internet, such as Arabic poetry, religious texts, or discussion forums? As far as we know, no current research effort exists in relation to these text genres.

2.2 Text preprocessing

Preprocessing is actually a trial to improve text classification by removing worthless information. It may include removal of numbers, punctuation (such as hyphens), and stop words, which are words that can be found in any text like prepositions and pronouns. In addition, Arabic texts need more consideration in this stage because of their writing style: (1) normalizing some writing forms that include Hamza "ء" and

Taa Marboutah “ة” to “ا” and 2) “ه”;) removing diacritics; and (3) removing kashida, a horizontal line that can be added in the middle of Arabic to certain letters as a form of justification. Most Arabic text classification takes into account the importance of preprocessing either fully or partially, but some research does not—see, for example, Sawaf et al. (2001) and Thabtah et al. (2008).

Because of the morphological nature of Arabic, some researchers consider root extraction and word stemming as a part of preprocessing (Kanaan et al. 2005; Syiam et al. 2006). In our opinion, using the full form of the word, its stem or root, is part of the feature extraction step, which will be discussed in Sect. 2.4.

2.3 Data division

After removing unwanted words and characters, the data are divided into two parts, training data and testing data. Based on training data, the classification algorithm will be trained to produce a classification model. The testing data will be used to assess the performance of the resulting classification model. Since there is no ideal ratio of training data to testing data, different ratios have been used for Arabic text classification research ranging from 25 % for training and 75 % for testing (Kanaan et al. 2005) up to 80 % for training and 20 % for testing (Sawaf et al. 2001).

The k-fold cross validation is sometimes used where different partitions for training and testing are used to produce k-classification models. The classification performance is the average performance of implemented classification models (see El-Halees 2008; Kanaan et al. 2009; Al-Saleem 2010).

2.4 Feature extraction

Texts are characterized by two types of features, external and internal. External features are not related to the content of the text, such as author name, publication date, author gender, and so on. Internal features reflect the text content and are mostly linguistics features, such as lexical items and grammatical categories. Most text classification research concentrates on the simplest of lexical features, the word. Using single words as a representative feature in text classification has proven effective for a number of applications (Diederich et al. 2003; Sebastiani 2002).

For Arabic text classification, words were treated as a feature on three levels: (1) using words in their orthographic form (Mesleh 2007; Thabtah et al. 2009); (2) word stems, in which the suffix and prefix were removed from the orthographic form of the word (Syiam et al. 2006; Kanaan et al. 2009); and (3) the word root, which is the primary lexical unit of a word (Elkourdi et al. 2004; Duwairi 2006). Whereas the above-mentioned methods focus on words as a way of reflecting meaning, another way is to focus on character n-grams, which usually convey no meaning. In this method, a certain number of consecutive characters are extracted and considered as features (Sawaf et al. 2001; Khreisat 2006). The output of this step is a list of features and their corresponding frequency in the training dataset.

2.5 Feature selection

The output of the feature extraction step is a long list of features, ranging from several thousand to hundreds of thousands. Not all of these features are beneficial for classification for several reasons: (1) The performance of some classification algorithms is negatively affected by the large number of features due to what is called curse of dimensionality. (2) An over-fitting problem may occur when the classification algorithm is trained in all features. (3) A large chunk of these features occur only once or twice in the training data. (4) Finally, some other features are common in all or most of the classes.

To overcome these problems, several methods were proposed to select the most representative features for each class in the training dataset. Feature selection methods statistically rank the features according to their distinctiveness for each class. Features with higher values are selected as the representative features. Different feature selection methods have been used in Arabic text classification. The most frequently used methods have been Chi Squared (CHI) (Syiam et al. 2006; Mesleh 2007; Thabtah et al. 2009; Zahran and Kanaan 2009); term frequency (TF), document frequency (DF) and their variations (Elksssourdi et al. 2004; Thabtah et al. 2008; Zahran and Kanaan 2009); and information gain (IG) (Syiam et al. 2006; El-Halees 2008). Apart from statistical ranking, word stems or roots were also used as feature selections where words with the same stem or root are considered as one feature, and features with higher frequency are used (Kanaan et al. 2005; Duwairi 2006; Bawaneh et al. 2008; Duwairi et al. 2009; Kanaan et al. 2009).

2.6 Data representation

In this step, the selected features from the previous step are formatted in a stable way to be represented to the classification algorithm. Usually, the data are represented as a matrix with n rows and m columns wherein the rows correspond to the texts in the training data, and the columns correspond to the selected feature. The value of each cell in this matrix represents the weight of the feature in the text. Several methods have been used to assign the proper weight to the feature. The most-used weighting methods have been term frequency inverse document frequency (TFiDF) (Syiam et al. 2006; Mesleh 2007; Bawaneh et al. 2008; Kanaan et al. 2009; Zahran and Kanaan 2009) and term frequency (TF) (Syiam et al. 2006; Kanaan et al. 2009).

2.7 Classification algorithm training and testing

In this step, the training matrix that contains the selected features and their corresponding weights in each text of the training data are used to train the classification algorithm. Classical machine learning algorithms have been the most used in Arabic text classification, such as Naïve Bayes (NB) (Elkourdi et al. 2004; Al-Saleem 2010); k-nearest neighbor (KNN) (Syiam et al. 2006; Bawaneh et al. 2008), and support vector machine (SVM) (Mesleh 2007; El-Halees 2008).

The training process yields a classification model that will be tested by means of the testing data. The same features that were extracted from the training data and the same weighting methods will be used to test the classification model.

2.8 Classification model evaluation

The ability of the classification model to classify texts into the correct classes results from all the previously described steps. A number of methods have been used to assess the performance of the classification model output, such as accuracy (Elkourdi et al. 2004; Bawaneh et al. 2008), precision and recall (Khreisat 2006; Kanaan et al. 2009), and f-measure (Syiam et al. 2006; Al-Saleem 2010).

From the data summarized in Table 1, it is difficult to suggest which combination of feature selection method, term weighting, and classification algorithm is the optimal solution for Arabic text classification because most of the datasets used are small and are mainly from the news genre. In the following sections, we will present our efforts on Arabic text classification as a follow-up to what we have discussed above.

3 Arabic text classification benchmarking dataset

One of the main objectives of this research is to build a benchmarking dataset (corpus) for Arabic text classification that takes into consideration corpus design criteria (Atkins et al. 1992; Sinclair 1995). The dataset design comprises seven sub-datasets covering different genres and subject domains. Each text in the corpus must be assigned to one of the defined classes. Table 2 illustrates the corpus genres, subject domains/classes, and number of texts for each class.

Table 2 King Abdulaziz city for science and technology corpus design

Genre	Classes	Total no. of texts
Saudi press agency	Cultural news, sports news, social news, economic news, political news, general news	1,500 texts evenly distributed
Saudi newspapers	Cultural news, sports news, social news, economic news, political news, general news, IT news	100 texts for each class from each newspaper. 4,200 texts in total. One newspaper per day
Websites	IT, economics, religion, news, medical, cultural, scientific	250 texts for each class. No more than 3 texts from each website
Writers	Ten writers	80 texts for each writer
Forums	IT, economics, religion, medical, cultural, scientific, sport, general	250 texts for each class. First 20 subjects from each discussion board.
Islamic topics	Hadeeth, aqeedah, lughah, tafseer, feqh	250 texts for each class
Arabic poems	Love, wisdom, description, praise, bemoaning, lampoon	250 texts for each class

Table 3 Statistical overview of compiled corpora

Genre	No. of classes	No. of texts	No. of words (tokens)	No. of unique words (Types)
Saudi press agency	6	1,526	253,472	36,497
Saudi newspapers	7	4,842	2,126,809	171,251
Websites	7	2,170	1,639,595	175,620
Writers	10	821	371,942	75,950
Forums	8	4,107	4,384,019	307,252
Islamic topics	5	2,243	2,463,442	286,589
Arabic poems	6	1,949	315,997	120,615
Total		17,658	11,555,276	

The datasets were assembled, comprising 17,658 texts, more than 11 million words, and seven different written genres—namely, the Saudi Press Agency (SPA), Saudi Newspapers (SNP), Websites, Writers, Forums, Islamic Topics and Arabic Poems. The Internet was the main venue used to collect the texts. A statistical overview of compiled corpora (genres) is shown in Table 3. Processing the component of this dataset and preparing it for classification algorithms is discussed in the next section.

4 Experiment automation

The benchmarking dataset illustrated in Sect. 3 needs to be processed according to text classification steps as mentioned in Sect. 2 and prepared in a suitable format for classification algorithms. A software tool called Arabic Text Classification tool (ATC tool) was developed in Java to handle and process the dataset. The user interface for the ATC tool is shown in Fig. 2.

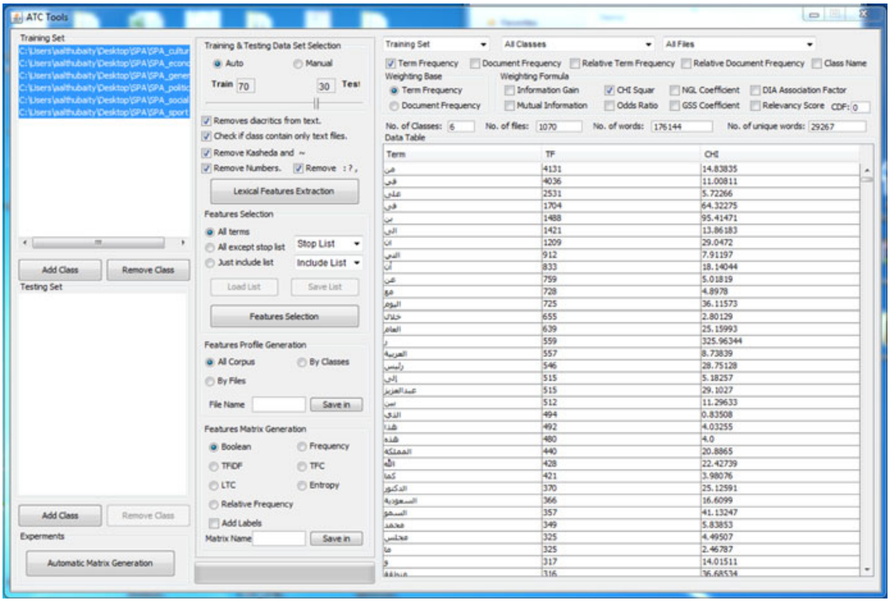
The ATC tool incorporates the following main functions:

(a) Text preprocessing: This allows the user to remove numbers, punctuations, kashida and stop words and to normalize the texts by removing diacritics.

(b) Data division: This divides the dataset into two sets - one for training and the other set for testing. The user can manually specify text files to be included in either sets. Alternatively, the software can randomly assign those text files to either training or testing sets based on user selection of how much percentage of the whole dataset each set (training/testing) is.

(c) Feature extraction: This extracts and generates the frequency list of the dataset features (single words). The function can list and save the features frequency for the whole dataset, for a specific class or file, or for training/testing sets; taking into consideration user selection mentioned earlier.. In addition, the user can explore the frequency profile for certain list of words. The document frequency, relative frequency and relative document frequency of features can also be explored and saved.

(d) Feature selection: This calculates the importance of each feature locally (for each class) and globally (for all classes) based on 10 feature-selection methods



ATC Tools

Training Set

C:\Users\aalthubaity\Desktop\SPA\SPA_cultur
C:\Users\aalthubaity\Desktop\SPA\SPA_econo
C:\Users\aalthubaity\Desktop\SPA\SPA_gener
C:\Users\aalthubaity\Desktop\SPA\SPA_politic
C:\Users\aalthubaity\Desktop\SPA\SPA_social
C:\Users\aalthubaity\Desktop\SPA\SPA_sport

Testing Set

Experments

Training & Testing Data Set Selection

● Auto ○ Manual

Train 70 30 Test

☑ Removes diacritics from text.
☑ Check if class contain only text files.
☑ Remove Kasheda and ~
☑ Remove Numbers. ☑ Remove : ? ,

Features Selection

● All terms
○ All except stop list: Stop List
○ Just include list: Include List

Features Profile Generation

● All Corpus ○ By Classes
○ By Files

File Name:

Features Matrix Generation

● Boolean ○ Frequency
○ TFIDF ○ TFC
○ LTC ○ Entropy
○ Relative Frequency
☐ Add Labels

Matrix Name:

Training Set | All Classes | All Files

☑ Term Frequency ☐ Document Frequency ☐ Relative Term Frequency ☐ Relative Document Frequency ☐ Class Name

Weighting Base
● Term Frequency
○ Document Frequency

Weighting Formula
☐ Information Gain ☑ CHI Squar ☐ NGL Coefficient ☐ DIA Association Factor
☐ Mutual Information ☐ Odds Ratio ☐ GSS Coefficient ☐ Relevancy Score CDF: 0

No. of Classes: 6 No. of files: 1070 No. of words: 176144 No. of unique words: 29267

Data Table

Term	TF	CHI
من	4131	14.83835
في	4036	11.00811
على	2531	5.72266
فى	1704	64.32275
بن	1488	95.41471
الى	1421	13.86183
ان	1209	29.0472
التي	912	7.91197
أن	833	18.14044
عن	759	5.01819
مع	728	4.8978
اليوم	725	36.11573
خلال	655	2.80129
العام	639	25.15993
ر	559	325.96344
العربية	557	8.73839
رئيس	546	28.75128
إلى	515	5.18257
عبدالعزيز	515	29.1027
بين	512	11.29633
الذى	494	0.83508
هذا	492	4.03255
هذه	480	4.0
المملكة	440	20.8865
الله	428	22.42739
كما	421	3.98076
الدكتور	370	25.12591
السعودية	366	16.6099
السمو	357	41.13247
محمد	349	5.83853
مجلس	325	4.49507
ما	325	2.46787
و	317	14.01511
منطقة	316	36.68534

Fig. 2 ATC Tool User Interface

- namely, term frequency (TF); document frequency (DF); information gain (IG); CHI squared (CHI); NG, Goh and Low (NGL) coefficient; Darmstadt indexing approach (DIA) association factor; mutual information (MI); odds ratio (OddsR); the Galavotti, Sebastiani, Simi (GSS) a coefficient and relevancy score (RS). The mathematical representations of these feature selection methods are illustrated in Table 4. The feature importance can be calculated based on term frequency or document frequency and can be explored according to their importance rank. Based on TF or DF threshold, these features can be filtered where certain features that are upper than certain threshold are considered only.

(e) Data representation This generates the training and testing matrix elements where each element represents one selected feature from previous step. The function weights the matrix elements according to seven weighting methods (Boolean, frequency, relative frequency, TFiDF, TFC, LTC, entropy). The mathematical representation of these methods is illustrated in Table 5. The software can automatically generate the training and testing matrices for multiple feature selection methods and multiple feature representation schemas based on document frequency and term frequency.

The resulting matrices are then used in other programs to build the classification model and to evaluate it. Those programs are RapidMiner 4.0 (Mierswa et al. 2006) and Clementine. RapidMiner is an open-source software which provides an implementation for all classification algorithms used in our experiments except the C5.0 algorithm. Clementine is a data-mining software from SPSS Inc. which provides an implementation for the C5.0 decision tree algorithm. The classification accuracy in

Table 4 The mathematical representation of feature selection methods

	Local	Global
TF	$TF(t,c_i) = F(t,c)$	$TF(t) = \sum_{i=1}^{i=m} F(t,c_i)$
DF	$DF(t,c_i) = D(t,c)$	$DF(t) = \sum_{i=1}^{i=m} D(t,c_i)$
IG	$IG(t,c_i) = \sum_{i=1}^{i=m} P(t,c_i) \cdot \log \frac{P(t,c_i)}{P(t).P(c_i)} + \sum_{i=1}^{i=m} P(\bar{t},c_i) \log \frac{P(\bar{t},c_i)}{P(\bar{t})\cdot P(c_i)}$	$IG(t) = -\sum_{i=1}^{i=m} P(c_i) \log P(c_i) + P(t) \sum_{i=1}^{i=m} P(t,c_i) \log P(t,c_i)$ $+P(\bar{t}) \sum_{i=1}^{i=m} P(\bar{t},c_i) \log P(\bar{t},c_i)$
CHI	$X^2(t,c) = \frac{T((P(t,c_i)\cdot P(\bar{t},\bar{c}_i) - P(t,\bar{c}_i)\cdot P(\bar{t},c_i)))^2}{P(t)\cdot P(\bar{t})\cdot P(c_i)\cdot P(\bar{c}_i)}$	$X^2(t) = \sum_{i=1}^{i=m} P(c_i) \cdot X^2(t,c_i)$
NGL	$NGL(t,c) = \frac{T\cdot((P(t,c_i)\cdot P(\bar{t},\bar{c}_i) - P(t,\bar{c}_i)\cdot P(\bar{t},c_i)))}{P(t)\cdot P(\bar{t})\cdot P(c_i)\cdot P(\bar{c}_i)}$	$NGL(t) = \sum_{i=1}^{i=m} P(c_i) \cdot NGL(t,c_i)$
DIA	$DIA(t,c_i) = \frac{P(t,c_i)}{P(t)}$	$DIA(t) = \sum_{i=1}^{i=m} P(c_i) \cdot DIA(t,c_i)$
MI	$MI(t,c_i) = \log \frac{P(t,c_i)}{P(t)\cdot P(c_i)}$	$MI(t) = \sum_{i=1}^{i=m} P(c_i) \cdot MI(t,c_i)$
OddsR	$OddsR(t,c_i) = \frac{P(t\|c_i).(1-P(t\|\bar{c}_i))}{P(t\|\bar{c}_i).(1-P(t\|c_i))}$	$OddsR(t) = \sum_{i=1}^{i=m} P(c_i) \cdot OR(t,c_i)$
GSS	$GSS(t,c_i) = P(t,c_i) \cdot P(\bar{t},\bar{c}_i) - P \cdot (t,\bar{c}_i) \cdot P(\bar{t},c_i)$	$GSS(t) = \sum_{i=1}^{i=m} P(c_i) \cdot GSS(t,c_i)$
RS	$RS(t,c_i) = \log \frac{P(t\|c_i)+d}{P(\bar{t}\|\bar{c}_i)+d}$	$RSS(t) = \sum_{i=1}^{i=m} P(c_i) \cdot RS(t,c_i)$

Where m is the number of classes; $F(t,c)$ is the number of times the term t occurs in class c_i; $D(t,c_i)$ is the number of documents in class c_i that contain the term t at least once; $P(t)$ is the probability of the term t; $P(c_i)$ is the probability of the class $c_{i;}$ $P(t,c_i)$ is the joint probability of the class c_i and the occurrence of the term t; T = Total number of documents in the corpus; $P(t|c_i)$ is the probability of t given c_i; d is a constant damping factor CDF

Table 5 The mathematical representation of feature representation methods

Function	Mathematical formula
BOOLEAN	$a = \begin{cases} 1 \ \textit{if the word exists in the text} \\ 0 \ \textit{if the word does not exist in the text} \end{cases}$
TF	$a = f(w)$
RF	$a = \frac{f(w)}{\sum_{i=1}^{i=n} f(w)}$
TFiDF	$a = f(w) \cdot \log\left(\frac{T}{d(w)}\right)$
TFC	$a = \frac{f(w) \cdot \log\left(\frac{T}{d(w)}\right)}{\sqrt{\sum_{i=1}^{i=n}\left[f(w) \cdot \log\left(\frac{T}{d(w)}\right)\right]^2}}$
LTC	$a = \frac{\log(f(w)+1) \cdot \log\left(\frac{T}{d(w)}\right)}{\sqrt{\sum_{i=1}^{i=n}\left[\log(f(w)+1) \cdot \log\left(\frac{T}{d(w)}\right)\right]^2}}$
ENTROPY	$a = \log(f(w)+1) \cdot \left(1 + \frac{1}{\log(T)} \sum_{i=1}^{i=T}\left[\frac{f(w)}{d(w)} \log\left(\frac{f(w)}{d(w)}\right)\right]\right)$

(w) equals the frequency of the word w in the text t; n equals total number of words in the text; $f(w)$ equals the frequency of the word w_i in the text t; T equals total number of texts in the data set and; $d(w)$ equals the number of texts t that the word w_i occurred in

the following experiments is computed by simply dividing the total number of correctly classified samples by the total number of samples in the testing dataset.

5 Assessing classification accuracy versus feature selection

This section aims to evaluate our basic classification methodology by employing frequently used classification algorithms: decision tree (C4.5), multilayer perceptron neural networks (MLP), support vector machines (SVM), Naïve Bayes (NB), and k-nearest neighbor (KNN). We ran the experiments on the SPA corpus which was divided into two distinct sets: training and testing. We selected two simple methods for term selection: TF (term frequency) and DF (document frequency). The top 10, 15, 20, 25, and 30 terms of each class in the corpus were selected as the representative terms, based on their related TF and DF. After we ranked the terms, the data were represented in two forms: Boolean and frequency.

To verify the effect of training data size on classification accuracy, we implemented three scenarios for each set of parameters: 30 % of corpus for training and the remaining 70 % for testing, 50 % of corpus for training and the remaining 50 % for testing and finally, 70 % of corpus for training and the remaining 30 % for testing. The classification accuracy of each scenario is shown in Table 6.

The NB algorithm shows the highest accuracy among all the five algorithms, 72.69 %. This rate was achieved using the top 30 terms in each class, with 70 % of the corpus used for training and the remaining 30 % for testing; term selection is based on document frequency and Boolean data representation. In all cases the best

Table 6 Classification accuracies for various classification algorithms

Classifier	Term per class	Boolean (%)						Frequency (%)					
		30		50		70		30		50		70	
		DF	TF	DF	TF	DF	TF	DF	TF	DF	TF	DF	TF
KNN	**10**	51.08	49.51	51.68	16.11	55.88	53.57	47.54	*	46.52	19.97	50.63	51.89
	15	50.30	52.07	49.61	49.36	53.36	56.09	44.19	46.65	43.94	*	52.31	51.89
	20	50.39	53.15	45.36	49.10	54.62	55.88	45.37	48.72	43.94	46.65	51.89	53.36
	25	*	52.17	48.20	48.58	53.36	57.14	44.49	46.26	43.30	45.49	52.10	53.99
	30	50.20	49.02	46.52	48.32	54.41	**58.19**	44.00	45.77	42.91	46.39	53.57	53.99
C4.5	**10**	56.00	56.59	53.48	26.03	61.97	54.83	59.55	*	53.87	28.74	61.13	52.94
	15	55.51	59.25	53.99	60.31	61.55	61.34	57.68	58.76	52.06	*	56.93	60.92
	20	55.91	59.06	57.35	55.80	62.61	63.03	56.20	57.48	56.57	52.06	61.97	**63.87**
	25	*	57.87	59.15	59.54	61.13	60.92	57.58	57.19	57.60	56.06	60.29	61.55
	30	57.97	57.87	55.93	60.82	57.56	61.13	57.58	58.07	60.82	61.08	60.29	61.34
NB	**10**	63.68	60.63	60.18	26.29	67.86	64.29	61.42	*	58.25	26.29	64.50	63.03
	15	65.26	65.55	62.89	62.63	68.70	68.07	62.70	62.89	62.24	*	65.97	66.81
	20	65.75	66.14	63.92	63.40	69.96	68.70	63.88	62.89	62.24	62.76	69.96	68.49
	25	*	67.91	65.85	65.34	70.17	70.17	64.76	64.47	64.95	64.69	69.96	69.96
	30	67.52	68.11	65.21	67.27	**72.69**	71.43	65.16	64.86	64.82	65.34	71.85	70.38
MLPs	**10**	57.87	56.99	54.38	19.46	55.04	56.30	54.82	*	52.45	20.23	57.98	53.57
	15	60.14	61.42	58.76	57.99	59.87	63.24	55.12	55.71	50.26	*	59.45	59.87
	20	61.12	61.52	58.51	61.47	65.55	62.18	52.46	58.37	51.03	54.90	60.50	63.24
	25	*	61.91	63.02	60.44	**67.65**	63.87	37.11	49.11	51.29	51.55	49.37	55.88
	30	62.50	63.29	60.05	61.60	63.45	64.71	50.69	36.32	46.65	30.67	60.71	61.34
SVM	**10**	60.93	59.06	54.38	15.85	62.61	60.92	56.1	*	51.16	17.91	60.29	58.61
	15	62.5	64.67	58.12	56.7	63.66	64.92	58.17	58.96	54.38	*	61.55	64.29
	20	62.5	64.67	59.92	59.92	69.12	67.02	59.06	59.74	55.28	54.51	66.18	64.5
	25	*	65.55	61.98	61.73	69.75	69.12	59.55	61.02	57.22	57.22	68.07	67.86
	30	63.58	65.75	61.86	63.02	69.75	**70.59**	61.22	61.42	58.38	57.86	68.07	65.97

Data in bold represents the maximum accuracy achieved

* Results are unavailable due to an error in the training and/or testing data

classification accuracy were achieved when the training data size is larger than testing data size.

Table 7 ranks the five classification algorithms according to their average accuracies. The next two columns of the table illustrate the highest accuracy rate for that classification algorithm and the equivalent experiment parameters: data representation, training set size, feature selection, and number of terms per class. The data illustrates the superiority of NB algorithm followed by SVM with average accuracy of 64.41 and 60.26 respectively. For all classification algorithms, the best classification accuracy achieved when Boolean representation is used except for C4.5 algorithm.

The top three classifiers ranked in Table 7: NB, SVM, and C4.5 were further evaluated using two more advanced methods for term selection: information gain (IG)

Table 7 Average accuracy and best accuracy for each classifier

Classifier	Average accuracy (%)	Best accuracy	
		Accuracy (%)	Dataset
NB	64.41	72.69	Boolean, 70, DF, 30
SVM	60.26	70.59	Boolean, 70, TF, 30
C4.5	57.28	63.87	Frequency, 70, TF, 20
MLPs	55.53	67.65	Boolean, 70, DF, 25
KNN	48.79	58.19	Boolean, 70, TF, 30

Table 8 Classification accuracy (%) using three classifiers and different term selection methods

Classifier	DF	IG	CHI	Average
SVM	68.86	71.71	72.15	70.91
NB	63.16	69.30	68.64	67.03
C4.5	62.28	65.79	65.57	64.55
Average	64.77	68.93	68.79	67.50

and CHI square (CHI). Table 8 shows the classification accuracy of those classifiers using three different term selection methods, and using all the other classification settings that yielded the best accuracy in the previous set of experiments. The IG and CHI weighting formulas were applied on document frequency. The training and testing sets were randomly compiled using the same corpus (SPA). Since the datasets were generated randomly for this experiment, the results of this experiment and of the previous experiment are not directly comparable. The SVM classifier shows the highest accuracy among the three classifiers, 72.15 % when CHI term selection method were used. This accuracy is very close to that NB achieved, 72.69 %, in Table 6.

Table 8 also ranks the average accuracy for the three classifiers. SVM also achieved the highest average accuracy at 70.91 %. Even though the highest accuracy was achieved using CHI square, the average accuracy of IG is slightly better than that of CHI square (68.93 % compared to 68.79 %). On the other hand, the least accurate results among the group were always associated with the DF term selection method.

We then studied the impact of the data representation schemes on the accuracy of the classification. Seven different representation schemes were used: relative frequency, entropy, LTC, TFC, TFiDF, frequency and Boolean. SVM was again implemented using the datasets used in the best case of Table 8. The results of this experiment are shown in Table 9. The best achieved accuracy remains the same as in Table 8 (72.15 %) using the Boolean representation scheme. The LTC scheme achieved an identical accuracy while the accuracy using relative frequency is very close (71.93 %). The least accurate results were with entropy (66.23 %).

Table 9 SVM classification accuracy using different representation schemes

Representation	Relative frequency	Entropy	LTC	TFC	TFiDF	Frequency	Boolean
Accuracy (%)	71.93	66.23	72.15	70.83	70.18	70.18	72.15

6 The impact of training and testing set size on classification accuracy

This set of experiments tested the current best settings from Tables 8, 9 on data from seven different corpora, including the SPA corpus. The classification settings used here are (classifier = SVM, representation = Boolean, training size = 70 and 90 %, term selection = CHI square, and terms = top 30/40/50 terms of each class). We used three stop word lists to filter out very common words from the data. A general stop word list was used with the following corpora (Writers, NP, Poems, and SPA). The forum stop word list was used with both the Web corpus, and the forums corpus. The third stop word list (the Islamic stop word list) was used with the Islamic Topics corpus.

Table 10 shows the results of this experiment using four runs. Run 1 is based on the best settings found in Tables 8, 9. The other runs show the effect on classification accuracy when the size of the training data and the number of terms per class increase.

In Run 1, the most accurate results were obtained using the Islamic Topics corpus (86.42 %). The Writers corpus comes next with an accuracy of 75.61 %. The classification accuracy decreased dramatically with the Arabic Poems corpus at 36.42 %. The classification accuracy using the remaining corpora is around 70 %. The average accuracy increased with each run, finally reaching 73.26 % after starting at 68.85 %, but the average in Run 4 showed little improvement over the average in Run 3 (0.14 % improvement). In all of the corpora except for Poems and SPA, individual accuracy improved with each run. The most noticeable result is from the Islamic corpus in Run 4 (accuracy of 95.05 %) and the result for the writers corpus (82.93 %) in the same run. On the other hand, there was an 18.76 % decrease in accuracy for the Poems corpus in Run 4.

Table 10 SVM and C5.0 classification results (%) using seven corpora and four runs

	Run	Training (%)	Testing (%)	Terms	SPA	SNP	Web	Writers	Forums	Islamic topics	Poems	Average
SVM	1	70	30	30	73.25	72.73	68.67	75.61	67.45	86.42	36.42	68.65
	2	90	10	30	76.67	73.43	72.09	74.39	62.99	88.29	39.49	69.62
	3	90	10	40	76.00	75.00	70.70	76.83	69.85	92.79	50.96	73.16
	4	90	10	50	73.33	75.20	76.28	82.93	68.63	95.05	41.40	73.26
C5.0	1	70	30	30	79.81	79.49	81.79	86.43	80.13	92.12	49.15	78.42
	2	90	10	30	80.96	80.84	81.88	84.98	80.18	92.38	47.04	78.32
	3	90	10	40	82.27	81.83	82.25	87.42	83.35	93.86	48.99	80.00
	4	90	10	50	82.92	83.55	83.21	86.74	82.67	93.96	50.52	80.51

We replicated the same set of runs but this time using C5.0 classifier as shown in Table 10. In Run 1, the most accurate results were obtained using the Islamic Topics corpus (92.12 %) as well as with the SVM classifier; however, the C5.0 Classifier gives better accuracy. The writers corpus comes next with an accuracy of 86.43 %. The classification accuracy decreased dramatically with the Arabic Poems corpus to reach 49.15 %. The average accuracy of Run 1 is 78.42 %. The average accuracy increased, run after run, reaching 80.51 % after starting at 68.85 %, but the average in Run 4 showed only a small improvement over the average in Run 3 (0.64 % improvement).

In all runs, the results of the C5.0 classifier overcame the results of the SVM classifier, excluding the Islamic corpus in Run 4. It was noticeable that the result for the Islamic corpus (accuracy of 95.05 %) is better than what was achieved with the C5.0 classifier (93.96 %). In addition, it was noticeable that in general the improvement in accuracy is minor over each run.

The orders of accuracies for both sets of experiments are of the same sort. However, the C5.0 classifier gives better results. This may be due to the splitting technique used with the C5.0 classifier. It works by splitting the sample, based on the field that provides the maximum information gain. Each subsample defined by the first split is then split again, usually based on a different field, and the process repeats until the subsamples cannot be split any further. Finally, the lowest level splits are re-examined, and those that do not contribute significantly to the value of the model are removed or pruned.

In Table 10, the Poems corpus yielded the lowest results among all the corpora. This is because of the nature of poetry, in which its quality highly relies on avoiding word repetition which, in turn, has a negative impact on the feature selection. When we excluded the Poems corpus, the average accuracy increased by almost 5 %.

7 Evaluating feature selection and feature representation

As previously illustrated, C5.0 and SVM algorithms produced more accurate classifications than the NB, C4.5, MLP, and KNN algorithms. A comparison between three term selection methods and seven data representation schemes is also reported. The CHI term selection method outperformed both the IG and DF methods, and the Boolean and LTC representation schemes were the most accurate schemes for classification. Additionally, the results revealed that increasing the number of selected terms improved the accuracy of the output. However, the results that were introduced earlier are based on a relatively small variation of datasets and can be further strengthened if similar experiments are applied on larger variations. Hence, the current experiment was designed to build on the previous experiments and to cover a wide variety of datasets.

In this experiment, classification accuracy was evaluated utilizing nine representation schemes and seven term selection methods, and using TF and DF as two different bases for term selection. Each corpus of the seven corpora was split into a training dataset (70 %) and a testing dataset (30 %). Each training dataset was used to generate 126 training matrices using all combinations of term selection methods and data representation schemes. All term selection methods have been set to select the top 200 terms from each class in the corresponding corpus. A total of 882 matrices were generated using the seven corpora. Common terms and words have all been filtered out using special stop word lists before applying term selection. The main classification algorithm used in this experiment is the SVM algorithm.

Table 11 shows the overall results of this set of experiments where each cell in the table illustrates the average classification accuracy for the seven corpora. The highest average accuracy is 80.53 %, which was achieved using TF as the term selection base with the GSS term selection method and the LTC representation scheme.

The following important findings are supported by the results:

(a) The MI and DIA term selection methods produced exactly the same results because they produced identical term rankings.
(b) Except for very few cases, OddsR also produced results similar to those of MI and DIA. The differences occurred in only six of 882 cases. This is less than 0.01 % of the total number of cases.
(c) Even though the highest average was achieved using TF, the overall average for DF (69.60 %) was slightly better than that of TF (68.16 %).
(d) The top eight most accurate results were achieved using the LTC representation scheme.
(e) The top three most accurate results were achieved using the TF term selection base.
(f) The top six most accurate results were achieved using either the GSS, RS, or None term selection methods.

Table 11 SVM classification results (%) using seven representation schemes and nine feature selection methods based on DF and TF

Base	Scheme	CHI	DIA	GSS	IG	MI	NGL	None	OddsR	RS
DF	Boolean	76.96	58.15	78.55	78.08	58.15	77.03	78.77	58.26	79.06
	Entropy*	73.93	58.03	74.72	75.11	58.03	74.00	62.14	53.28	62.25
	Frequency	73.10	57.05	75.31	73.54	57.05	73.38	74.28	53.29	74.53
	LTC	78.52	58.55	80.05	79.23	58.55	78.56	80.15	58.55	80.20
	R. freq.	73.74	58.03	75.18	73.78	58.03	74.00	74.56	54.40	74.18
	TFC	76.76	58.28	78.79	77.77	58.28	76.85	78.78	58.47	78.68
	TFiDF	75.31	57.29	76.90	76.33	57.29	75.42	76.18	57.33	76.06
TF	Boolean	77.73	57.98	78.98	78.22	57.98	77.48	78.28	57.98	78.54
	Entropy	61.63	48.84	62.01	61.82	48.84	61.63	61.54	48.84	61.40
	Frequency	72.20	55.60	74.65	73.33	55.60	72.44	74.71	55.60	74.78
	LTC	78.93	58.16	80.53	79.63	58.16	78.83	80.33	58.16	80.39
	R. freq.	73.10	57.74	74.35	73.10	57.74	73.30	74.23	57.74	74.12
	TFC	76.78	57.60	78.94	77.33	57.60	76.68	78.84	57.60	78.66
	TFiDF	74.53	55.12	75.95	75.18	55.12	74.44	75.68	55.12	75.56

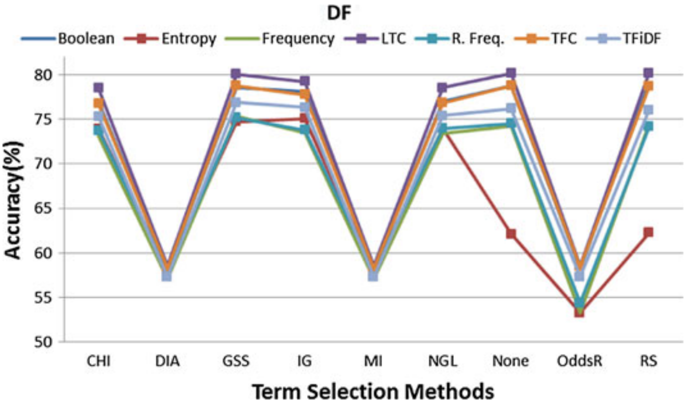


Fig. 3 Experimental results using DF as the base for the term selection

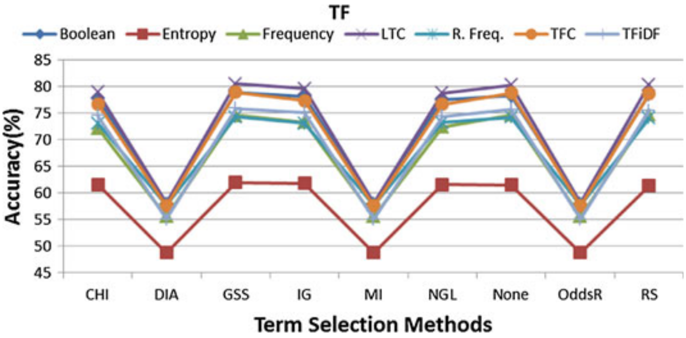


Fig. 4 Experimental results using TF as the base for the term selection

(g) LTC always produced the highest accuracy with all the term selection methods used in this experiment, as shown in Figs. 3 and 4, followed by Boolean and TFC.

(h) Entropy seems to work better with DF than TF.

(i) Based on the top ten average accuracies extracted from Table 11, Table 12 illustrates the classification accuracies for each corpus using the combinations representing those top ten average accuracies. There were no significant differences between the top 10 averages in this experiment; the difference between the highest average and the 10th highest average was only 1.55 %, as shown in Table 12.

Table 13 presents the classification accuracy for each corpus in the set. The numbers are shown in two main columns. The first column shows the accuracy as it occurs in the best overall average (TF-LTC-GSS), while the second column shows the best accuracy of each corpus using different methods. The main aspects of these results are summarized below:

Table 12 Classification accuracy details for the top ten best average accuracy experiment

Corpus	TF-LTC-GSS	TF-LTC-RS	TF-LTC-None	DF-LTC-RS	DF-LTC-None	DF-LTC-GSS	TF-LTC-IG	DF-LTC-IG	DF-Boolean-RS	TF-Boolean-GSS
SPA	77.85	77.85	75.88	76.97	76.97	76.32	74.78	77.19	74.78	75.88
SNP	78.47	78.28	79.19	78.47	78.34	78.73	78.67	77.95	76.91	77.10
Websites	87.81	87.50	87.04	86.57	86.57	87.19	87.04	85.96	83.64	84.57
Writers	86.59	86.59	87.40	89.02	89.02	87.40	87.80	86.59	93.50	92.28
Forums	81.77	81.04	81.04	81.37	81.69	81.69	79.90	78.11	79.98	78.44
Islamic topics	95.67	96.12	95.52	95.52	95.82	95.37	95.97	94.93	92.84	93.88
Arabic poems	55.58	55.37	56.21	53.47	52.63	53.68	53.26	53.89	51.79	50.74
Average	80.53	80.39	80.33	80.20	80.15	80.05	79.63	79.23	79.06	78.98

Table 13 Ranking the seven corpora based on their average accuracies

Corpus	Best overall average (TF-LTC-GSS)	Best result per corpus	
	Accuracy (%)	Accuracy (%)	Methods
Islamic Topics	95.67	96.12	TF-LTC-RS, TF-TFC-RS
Web	87.81	87.81	TF-LTC-GSS
Writers	86.59	93.50	DF-Boolean-None, DF-Boolean-RS
Forums	81.77	81.77	TF-LTC-GSS
SNP	78.47	79.19	TF-LTC-None
SPA	77.85	78.29	TF-TFC-None
Arabic poems	55.58	56.84	TF-TFC-GSS

(a) Except for the Writers corpus, accuracies in the best overall average are equal or very close to the accuracies in the best result cases.

(b) The difference between the accuracies in the two columns in the Writers row is significant (about 7 %). The Writers corpus seems to work better with a Boolean representation scheme than with any other representation scheme. The top six results for this corpus all used the Boolean scheme, indicating that some words are very important in revealing the identity of the writers in this corpus, regardless of how many times these words occurred in each article.

(c) Except for the Poems corpus, the accuracy associated with each corpus ranges from good to excellent.

(d) The best achieved accuracy for the Arabic Poems corpus is 56.84 %, representing a very poor performance compared to other corpora in the experiment. This result is attributed mainly to the principles of writing poems in general, and especially for writing Arabic poems. Poem writing is a very creative form of writing, so authors tend to select and invent their own vocabulary. This tendency can be seen very clearly if we look at the percentage of the types (unique words) and total tokens (words) in this corpus. This percentage equals about 38 %, which is the highest in this experiment and which indicates that, on average, each word occurs only 2.6 times in the corpus. This fact will eventually result in a poor classification performance.

(e) Additionally, poem writing involves a variety of writing techniques, such as metaphor and symbolism, which make the classification process more difficult.

SVM is the main classification algorithm in this section. It showed very good results however in order to evaluate other classification algorithms that showed promising results in previous experiments, we ran the same experiment using the NB and C4.5 algorithms and then compared results. The results of this comparison are shown in Table 14. The SVM algorithm outperformed the other two classification algorithms, with an average improvement of 6.56 % over NB and 31.58 % over C4.5. The SVM results were achieved using TF as the base for term selection, GSS as the term selection method, and LTC as the representation scheme. The TFC and None term selection methods also produced very good results.

8 The impact of number of features on classification accuracy

The results from the previous experiments helped us identify the method that gave the best average performance for classification—i.e., TF-LTC-GSS using the SVM classification algorithm. In this experiment, we tried to determine if there is still room for further improvement by using more terms. Results from previous experiments indicated that the use of more terms will probably improve the performance.

Table 15 presents the results using a different number of terms. Terms were selected as being in the top 1, 2, … 20 % terms of each class in the related corpus using the GSS feature selection function. The results in general indicate that we obtained better accuracies with higher percentages of the number of terms, but improvements in some cases were not significant. With four of the seven corpora, it was not possible to run the experiment with higher percentages of the number of terms because of memory size limitations. Figure 5 illustrates the results graphically.

The average improvement that occurred when increasing the number of terms from 1 % to at most 20 % was 7.17 %. The greatest improvement in accuracy was recorded for the Poems corpus (49.68–60.63 %). In contrast, the Islamic Topics corpus exhibited the least improvement in accuracy (96.12–96.72 %). We concluded, therefore, that further improvement in accuracy can be achieved by increasing the number of terms. The factors that govern the choice of more terms involve the available memory resources and speed requirements. If the available memory is limited and classification speed is a concern, then we recommend using fewer terms for the analysis.

Table 14 Comparison of results for SVM, NB, and C4.5

Corpus	SVM			NB			C4.5		
	TF-LTC-GSS	Best result		DF-LTC-GSS	Best result		DF-LTC-CHI	Best result	
Islamic topics	95.67	96.12	TF-LTC-RS TF-TFC-RS	90.60	91.04	TF-LTC-IG	81.79	82.84	DF-LTC-NGL
Websites	87.81	87.81	TF-LTC-GSS	81.17	82.72	TF-Freq-CHI TF-Freq-NGL	65.74	68.52	TF-Bool-IG
Writers	86.59	93.50	DF-Bool-None DF-Bool-RS	78.05	78.05	DF_LTC-GSS	49.59	52.85	DF-RFreq-CHI DF-RFreq-NGL
Forums	81.77	81.77	TF-LTC-GSS	74.29	74.29	DF-Bool-CHI DF-LTC-GSS TF-LTC-GSS	59.24	64.04	DF-Freq-IG
NP	78.47	79.19	TF-LTC-None	74.69	75.34	DF-TFC-GSS	67.58	67.58	TF-LTC-CHI
SPA	77.85	78.29	TF-TFC-None	75.44	76.97	TF-LTC-GSS TF-LTC-IG TF-TFC-GSS	64.04	66.23	DF-Bool-CHI
Arabic poems	55.58	56.84	TF-TFC-GSS	54.74	56.42	DF-Freq-GSS	40.42	41.89	TF-RFreq-CHI

Table 15 Classification accuracy using more terms

Corpus	No. of classes	No. of texts	Description	Percentage of the number of terms per class						
				1	2	5	7	10	15	20
Writers	10	821	Accuracy	87.80	92.28	92.28	93.50	94.31	95.53	95.93
			No. of terms	976	1,952	4,879	6,829	9,756	14,634	19,510
Forums	8	4,107	Accuracy	84.62	89.67	*	*	*	*	*
			No. of terms	4,065	8,229	20,377	28,527	40,753	60,533	80,713
Arabic poems	6	1,949	Accuracy	49.68	52.84	57.68	58.53	55.79	60.42	60.63
			No. of terms	1,017	2,033	5,079	7,111	10,160	15,236	20,315
SPA	6	1,526	Accuracy	76.54	79.17	80.04	80.48	80.48	80.70	80.70
			No. of terms	492	987	2,463	3,448	4,926	7,388	9,851
SNP	7	4,842	Accuracy	81.15	82.13	*	*	*	*	*
			No. of terms	2,717	5,435	13,654	19,118	27,376	41,065	54,751
Websites	7	2,170	Accuracy	84.88	86.57	88.12	89.04	89.66	*	*
			No. of terms	2,251	4,500	11,249	15,745	22,492	34,141	46,432
Islamic topics	5	2,243	Accuracy	96.12	95.97	96.57	96.57	96.72	*	*
			No. of terms	1,817	3,631	9,076	12,707	18,152	27118	36157

* Out-of-memory on a PC with 2 GB RAM

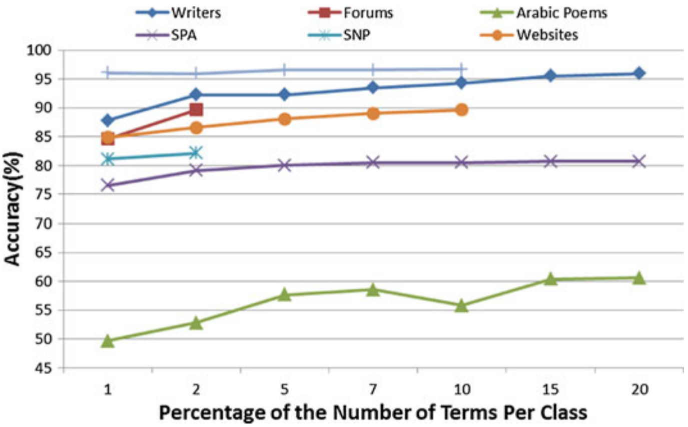


Fig. 5 Classification performance using more terms

9 Conclusion

In addition to building large Arabic corpora for text classification, the main contribution of this paper was to investigate a variety of text classification techniques using the same datasets. These techniques include a wide range of classification algorithms, term selection methods, and representation schemes. The classification techniques used in this paper have been widely used by many researchers for the same task. However, to the best of our knowledge, none of the previous works has tried to compare the accuracy of all of these techniques when applied to datasets that belong to a large spectrum of genres, as presented in this paper.

Several classification algorithms were tested in this study (C4.5, C5.0, MLP neural networks, SVM, NB, and KNN algorithms). SVM produced the most accurate classification in the main experiments presented in this paper. The next most noteworthy classification algorithms were C4.5 and NB. However, SVM showed much better results than the other two algorithms, outperforming NB, on average, by 6.56 % and C4.5 by 31.58 %. Some experiments were conducted using the C5.0 decision tree algorithm. In these experiments, C5.0 produced outstanding results that outperformed those from SVM. For term selection, we compared several methods used frequently in the literature. The investigated methods were CHI, DIA, GSS, IG, MI, NGL, None, Odds ratio, and RS. The None method involved using either TF or DF as the only base for term ranking. GSS, None, and RS were the three methods that showed the best results. Our best average result was achieved using the GSS method with TF as the base for calculations.

Several representation schemes, also known as term weighting functions, were evaluated in addition. These included Boolean, frequency, LTC, TFiDF, TFC, entropy, and relative frequency. The experimental results showed that LTC was superior, followed by Boolean and TFC. A related issue in term selection is the proper selection of the required number of terms to achieve good classification accuracy. The results demonstrated that a higher number of terms produced better accuracy, although the improvements saturate after a certain limit. Factors that govern the choice of the number of terms are related to memory and speed—i.e., how much memory is available and how fast the classification process should be.

The overall results of the different experiments presented in this paper are very good except for the poems corpus. The average would be way better without this one. The best classification accuracy ranges from 60.63 to 96.72 % using seven corpora, representing an average of 85.06 %. The accuracy differs greatly between corpora. The corpus with the most accurate result was the Islamic Topics corpus, while the Arabic Poems corpus yielded the least accurate result. Future work will consider other issues related to Arabic text classification. These include employing linguistic information such as word stems and parts of speech. This approach should be attainable, given the current increased interest in Arabic Natural Language Processing (NLP) in the research community.

Acknowledgments This project was fully funded by King Abdulaziz City for Science and Technology via grant number 104-27-30. The authors would like to thank the two anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

Al-Saleem, S. (2010). Associative classification to categorize Arabic data sets. *The International Journal Of ACM JORDAN, 1*, 118–127.

Atkins, S., Clear, J., & Ostler, N. (1992). Corpus design criteria. *Literary and Linguistic Computing, 7*(1), 1–16.

Bawaneh, J. M., Alkoffash, M. S., & Alrabea, A. I. (2008). Arabic text classification using K-NN and Naive Bayes. *Journal of Computer Science, 4*, 600–605.

Diederich, J., Kindermann, J. L., Leopold, E., & PAAß, G. (2003). Authorship attribution with support vector machines. *Applied Intelligence, 19*(1/2), 109–123.

Duwairi, R. (2006). Machine learning for Arabic text categorization. *Journal of the American Society for Information Science and Technology JASIST, 57*(8), 1005–1010.

Duwairi, R., Al-Refai, M., & Khasawneh, N. (2009). Feature reduction techniques for Arabic text categorization. *Journal of the American Society for Information Science, 60*(11), 2347–2352.

El-Halees, A. (2008). A comparative study on Arabic text classification, *Egyptian Computer Science Journal, 30*(2). http://www.informatik.uni-trier.de/~ley/db/journals/ecs/ecs30.html

Elkourdi, M., Bensaid, A., & Rachidi, T. (2004). Automatic Arabic document categorization based on the Naive Bayes algorithm. In *Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-Based Languages*, (pp. 51–58).

Kanaan, G., Al-Shalabi R., & Al-Azzam, O. (2005). Automatic text classification using Naïve Bayesian algorithm on Arabic language. In *Proceedings of the 5th International Business Information Management Conference (IBIMA),* (pp. 327–339).

Kanaan, G., Al-Shalabi, R., Ghwanmeh, S., & Al-Ma'adeed, H. (2009). A comparison of text-classification techniques applied to Arabic text. *Journal of the American Society for Information Science and Technology, 60*(9), 1836–1844.

Khreisat, L. (2006). Arabic text classification using N-gram frequency statistics a comparative study. In *Proceedings of the 2006 International Conference on Data Mining*, (pp. 78–82).

Mesleh, A. A. (2007). Chi square feature extraction based Svms Arabic language text categorization system. *Journal of Computer Science, 3*(6), 430–435.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). YALE: Rapid prototyping for complex data mining tasks. In L. Ungar, M. Craven, D. Gunopulos, & T. Eliassi-Rad (Eds.), *KDD 06 Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 935—940*. New York, USA: ACM.

Sawaf, H., Zaplo, J., & Ney, H. (2001). Statistical classification methods for Arabic news articles. *Arabic Natural Language Processing Workshop, ACL'2001,* (pp. 127–132).

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34* (1), 1–47.

Sinclair, J. (1995). Corpus typology–a framework for classification. In G. Melchers & B. Warren (Eds.), *Studies in anglistics* (pp. 17–33). Stockholm: Almqvist & Wiksell.

Syiam, M. M., Fayed, Z. T., & Habib, M. B. (2006). An intelligent system for Arabic text categorization. *International Journal of Intelligent Computing and Information Sciences, 6*(1), 1–19.

Thabtah, F., Eljinini, M., Zamzeer, M., & Hadi, W. (2009). Naïve Bayesian based on Chi Square to categorize Arabic data. In *Proceedings of The 11th International Business Information Management Association Conference (IBIMA) Conference on Innovation and Knowledge Management in Twin Track Economies*, (pp. 930–935).

Thabtah, F., Hadi, W., & Al-Shammare, G. (2008). VSMs with K-Nearest Neighbour to categorise Arabic text data. In *The World Congress on Engineering and Computer Science 2008,* (pp. 778–781).

Zahran, M. M., Kanaan, G., & Habib, M. B. (2009). Text feature selection using particle Swarm optimization algorithm. *World Applied Sciences Journal, 7*(Special Issue of Computer & IT), 69–74.