International Journal of Computational Intelligence and Applications
Vol. 8, No. 3 (2009) 315–343

Imperial College Press
www.icpress.co.uk

ARABIC SCRIPT WEB PAGE LANGUAGE IDENTIFICATION USING HYBRID-KNN METHOD

ALI SELAMAT*, IMAM MUCH IBNU SUBROTO†
and CHOON-CHING NG‡

Intelligent Software Engineering Laboratory
Faculty of Computer Science & Information Systems
University of Technology Malaysia
81310 UTM Skudai, Johor, Malaysia
**aselamat@utm.my*
†*imam_mis@yahoo.com*
‡*choonching5u@gmail.com*

Revised 5 June 2009

In this paper, we proposed hybrid-KNN methods on the Arabic script web page language identification. One of the crucial tasks in the text-based language identification that utilizes the same script is how to produce reliable features and how to deal with the huge number of languages in the world. Specifically, it has involved the issue of feature representation, feature selection, identification performance, retrieval performance, and noise tolerance performance. Therefore, there are a number of methods that have been evaluated in this work; k-nearest neighbor (KNN), support vector machine (SVM), backpropagation neural networks (BPNN), hybrid KNN-SVM, and KNN-BPNN, in order to justify the capability of the state-of-the-art methods. KNN is prominent in data clustering or data filtering, SVM and BPNN are well known in supervised classification, and we have proposed hybrid-KNN for noise removal on web page language identification. We have used the standard measurements which are accuracy, precision, recall and $F1$ measurements to evaluate the effectiveness of the proposed hybrid-KNN. From the experiment, we have observed that BPNN is able to produce precise identification if the data set given is clean. However, when increasing the level of noise in the training data, KNN-SVM performs better than KNN-BPNN against the misclassification data, even on the level of 50% noise. Therefore, it is proven that KNN-SVM produce promising identification performance, in which KNN is able to reduce the noise in the data set and SVM is reliable in the language identification.

Keywords: Arabic script language identifications; support vector machine (SVM); backpropagation neural networks (BPNN); k-nearest neighbors (KNN); KNN-SVM; KNN-BPNN; hybrid-KNN.

1. Introduction

Language is a term used to refer to the natural language used for human communication either in spoken or written forms. These are 7,000 languages that have been reported in *Ethnologue*, a widely cited reference for languages around the world.[1]

*Corresponding author.

Globalization has led to unlimited information sharing across the Internet, where communication among people in a bilingual environment is a critical problem to be overcome. Abd Rozan *et al.* (2005)[2] have justified the importance of monitoring the behavior and activities of world languages in cyberspace. The information collected from such a study has implications on customized education, in which Information and Communication Technology (ICT) has to cope with the "digital divides" that exist both within countries and regions, and between countries.[a] Furthermore, they also found that the ubiquitous learning process (learning present everywhere at once) is better conducted with a native language. In addition, Maclean[3] has reasserted the status of language as a topic of major interest to researchers in the light of the rise of the transnational corporation. Also, Redondo-Bellon (1999)[4] has also analyzed the effects of bilingualism on the consumer in Spain. All these examples reflect the importance of multi-languages in globalization. According to the book *The World is Flat* by Friedman (2005),[5] the author writes:

> "The net result of this convergence was the creation of a global, Web-enabled playing field that allows for multiple forms of collaboration-the sharing of knowledge and work-in real time, without regard to geography, distance, or, in the near future, even language. No, not everyone has access yet to this platform, this playing field, but it is open today to more people in more places on more days in more ways than anything like it ever before in the history of the world. This is what I mean when I say the world has been flattened."

According to Internet World Stats, the Internet usage increased dramatically between 2000 and 2008 in the world, for example in Middle Eastern countries such as Iran, Syria, Saudi Arabia, Yemen, etc.[6,7] In addition, the Summer Institute of Linguistics has reported that there are 69 languages spoken or used by more than 10 million people in the world, including English.[8] Since there are many people such as Japanese, Arabic, Chinese, etc., that do not use an international language like English, therefore language identification is needed to support a multilingual processing system. Language identification is the process of determining the predefined language automatically for the given content (e.g., English, Malay, Chinese, Japanese, Arabic, etc.). In various applications, language is an important tool for human communication and presently, the language dominating the Internet is English. A web page is a kind of digital document displayed in a web browser. The web page can be written using diverse languages or different scripts of encoding scheme such as Unicode.[9] One of the crucial tasks in identifying the language is that same words may appear in many languages which use the same scripts. This usually happens when these countries are using Arabic scripts for their written language. Therefore, in this paper we revisit the problem of Arabic script web page

[a]Digital divide refers to the disparity between those who have use of and access to ICT versus those who do not.[2]

language identification by proposing hybrid-KNN methods. Initially, the KNN-SVM has been proposed for web page language identification.[10] In this work, the comparison with BPNN and KNN-BPNN in terms of identification performance, retrieval performance and noise tolerance performance is further described. KNN has been used to find out the best features from the data sets by removing the outliers. It has been proven capable in data filtering.[11–13] Then, the features are fed into SVM or BPNN for language identification since both methods are powerful techniques in pattern recognition. SVM is capable in dealing with high dimensional data and BPNN is good for learning complex mapping between input and outputs.[14–16] Therefore, both methods have been chosen as identifiers due to the fact that text-based language identification consisting of up to thousands of classes. There are many noises that may exist in the text that can directly affect the performance of language identification, so the hybrid-KNN is proposed to evaluate the effectiveness of web page language identification against noise.

This paper is organized as follows: Related works of language identification is discussed in Sec. 2. The proposed hybrid-KNN methods and its conventional methods are described in Sec. 3. Section 4 explains the preprocessing and evaluation measurements. Experimental results such as identification performance, retrieval performance and noise tolerance performance are discussed in Sec. 5. Finally, the discussions and conclusions are presented in Secs. 6 and 7, respectively.

2. Related Works

A practical identifier usually can produce higher identification accuracy with low computational memory and shorter processing time. Mislabeled training documents will also affect the results of language identification.[17] Sibun and Reynar (1996)[18] have stated that language identification factors need to be taken into consideration, including the type of features to be used, the dimension of the data sets, and the type of analysis to be used in validating the language identification results. Botha *et al.* (2006)[19] stated that accuracy of web page language identification depends on a number of factors, including the size of the textual fragment, the amount and variety of training data, the classification algorithm employed and the similarity of the languages to be discriminated. In general, problems existing in web pages include irrelevant information, unstructured information, spelling or syntax errors, and an overabundance of international terms.[20–22] For example, when we encounter a word *main*, we do not know if it is an English word (referring to "most important") or Malay word (referring to a word "play"). Biemann and Teresniak (2005)[23] argue that supervised training has a major drawback, in which the language identifier will fail on languages that are not contained in its training. As it will for the most part have no clue about that, it will assign some arbitrary or unknown language.

Web page language identification has received less attention than spoken language identification, as it is argued that this is a straightforward task.[18] However, Xafopoulos *et al.* (2004)[24] and Hughes *et al.* (2006)[25] argue that web page language

identification has a number of questions which remain open and ripe for further investigation. For example, the impact of preprocessing, minority language identification, multilingual identification, supervised or unsupervised identification and features processing.

Feature selection has at its function to reduce unnecessary attributes of the original content. It not only cuts down the loads of learning algorithm, but also reduces bias in raw data and increases the effect of learning result. Several feature selection methods have been proposed in the literature. For example, entropy, small word technique, Unicode based identification, web page information, Principle Component Analysis (PCA), etc.[26–29] The language identification problem can be seen as an instance of a more general problems that of classifying objects using attributes. For this purpose different kinds of attributes have been used. For example, character,[30] word,[31–33] word classes,[34] particular n-grams,[35–37] sentences,[23] etc. Many approaches have been developed for written language identification such as vector space modeling,[16] neural network,[38–41] statistical approaches,[42,43] and support vector machines.[27]

With the rapid emergence and explosion of the Internet and the trend of globalization, a tremendous number of web pages written in different languages are electronically accessible online. Efficient and effective management of these web pages written in different languages is important to organizations and individuals. For this purpose, many studies have been done in order to identify automatically the language in which the information is written on a web page.[24] A suitable method of feature selection or extraction of web pages is required to extricate the useful features from web pages before an identification process is done. Indirectly, the classification performance can be increased if the features used are reliable and robust.[19] Isa *et al.* (2009)[44] have proposed a hybrid approach to classify the documents based on the self organizing map (SOM) and Bayes formula. However, the approach focuses only on English text documents. Saeed and Albakoor (2009)[45] have analyzed the applicability of neural networks for typewritten and handwritten text recognition. The authors have used the segmentation algorithms to support the detection of the slope within the images of handwritten Arabic scripts. However, comparative studies between the k-nearest neighbor and neural networks applied on the texts have not been analyzed by the authors. Fattah and Ren (2009)[46] have done comparative studies of language summarizations using feed forward neural network (FFNN), mathematical regression (MR), probabilistic neural network (PNN) and Gaussian mixture model (GMM) in order to construct a text summarizer on Arabic and English texts. However, the extensive comparison on the usage of k-nearest neighbor for sentence summarizations has not been explored in the paper. Wang *et al.* (2009)[47] have identified the theme logic model (TLM) in order to present all the themes in a text and the logical relations of different themes to be used as the input to neural networks to acquire knowledge from failure analysis reports. However, the analysis on knowledge acquisition is mainly in English texts. The identification of the failure analysis on the Arabic texts has not been

studied by the authors. Much efforts have been made to prevent the fall-out in using minority languages in the online community and less-computerized languages. With the increasing number of web pages on the Internet, it has become a necessity to provide some techniques to identify and retrieve effectively encoded information automatically.

3. Proposed Hybrid-KNN Methods

In this section, we discuss the details algorithm of five methods that have been utilized in this work for Arabic script web page language identification. There are k-nearest neighbor (KNN), support vector machine (SVM), backpropagation neural networks (BPNN), hybrid KNN-SVM and KNN-BPNN (as shown in Fig. 1).

3.1. *K-Nearest Neighbor (KNN)*

A k-nearest neighbor (KNN) classifier determines the class label of a test example based on its k neighbors that are close to it. Any test example is classified into the class that has the most number of examples among its k closest neighbors. Among all classifier algorithms, k-nearest neighbor is widely used as a text classifier because of its simplicity and efficiency.[48] Figure 2 shows the KNN classifier with $k = 3$. The classifier calculates the three nearest data samples and predicts the class of the document closest to it. The Euclidean distance formula, as stated in Eq. (1), has been effectively used to calculate the distance among neighbors. Referring to Fig. 2, three examples of web documents x_1, x_2 and x_3 will be used to predict the language that they belong to by using the KNN classifier. The KNN classifier has been able to

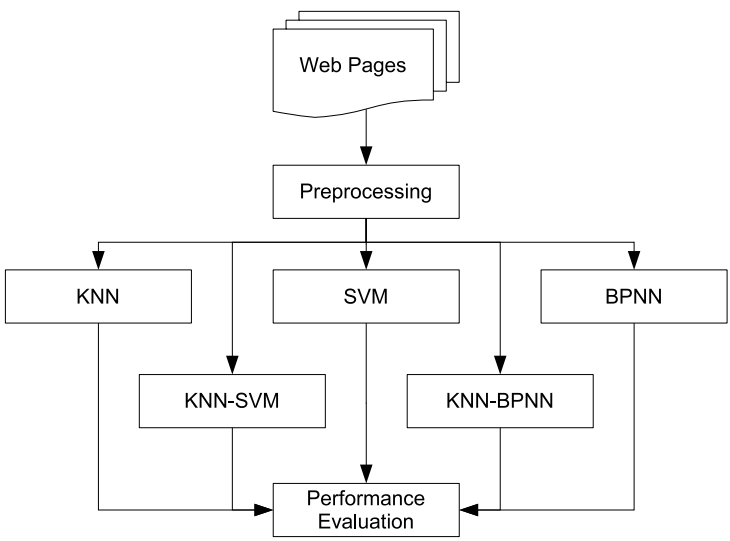

Fig. 1. The flow overview of the Arabic script web page language identification.

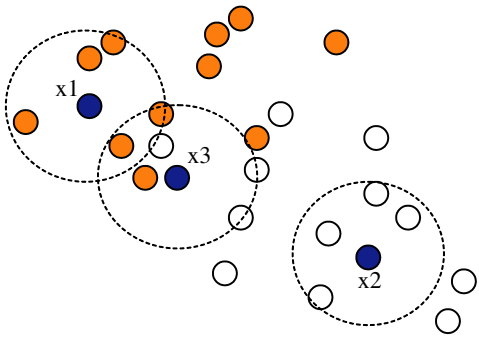

Fig. 2. A k-nearest neighbor (KNN) classifier.

predict that document x_1 belongs to a positive (+) class and document x_2 belongs to a negative (−) class and document x_3 is predicted to be in a positive (+) class.

$$Distance_{AB} = \sqrt{\sum_{i=1}^{n} (x_{A,i} - x_{B,i})^2}. \tag{1}$$

3.2. *Support Vector Machine (SVM)*

A support vector machine (SVM) is a relatively new statistical classification method proposed by Vapnik in 1995.[49] Based on the structural risk minimization (SRM) principle, the SVM tries to find a separating hyperplane with maximum margins to separate the positive examples and negative examples from the training data sets. It makes decisions based on the support vectors that are selected as the only effective elements from the training set.

In the learning stage, the SVM finds the parameters $w = [w_1 w_2 \cdots w_n]^T$ and b of discriminant or decision function $d(x, w, b)$ from the training data sets as follows:

$$d(x, w, b) = w^T + b = \sum_{i=1}^{n} w_i x_i + b. \tag{2}$$

Figure 3 shows that the SVMs finding the hyperplane h, which is separated from the positive and negative training examples with a maximum margin. The examples that are close to the hyperplane are called support vectors, which are marked with a circle.

Let $\{x_1, \ldots, x_n\}$ be the data set and let $y_i \in \{1, -1\}$ be the class label of x_i. The decision boundary should classify all points correctly:

$$y_i(w^T + b) \geq 1, \quad \forall\, i. \tag{3}$$

The decision boundary can be found by solving the following constrained optimization problem:

$$minimize\ \frac{1}{2}\|w\|^2, \quad Subject\ to\ y_i(w^T + b) \geq 1. \tag{4}$$

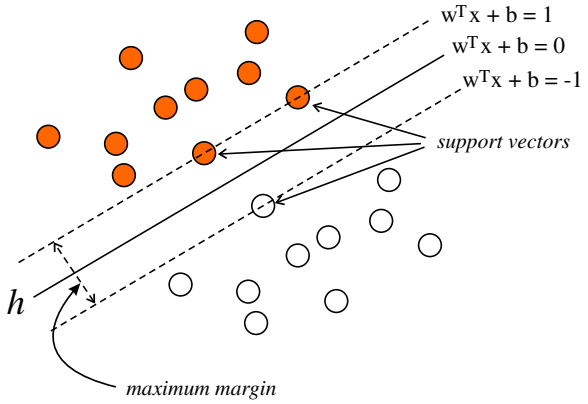

Fig. 3. Support vector machines (SVM) find the hyperplane.

The SVM classifier has been designed for binary classification that clearly separated the positive and negative classes from the tested data sets. As the Arabic script language identification is based on a multi-class of problems that involve Arabic, Persian, Urdu and Jawi languages, we have divided them into two groups as the positive and negative classes. The Arabic data set has been marked as a positive (+) class and the others are marked as negative (−) classes. This convention has been applied to other languages as well. Therefore, four SVM classifiers that correspond to four languages will be used in Arabic script language identification. The SVM classifier tool SVM-light (developed by Thorsten Joachims[50]) was used in our experiments.

3.3. *Backpropagation Neural Networks (BPNN)*

Artificial Neural Network (ANN) is fundamentally a parallel processor. ANNs are computer programs that are biologically inspired to simulate the way in which the human brain processes information. It has been applied to many applications including language identification because of their fascinating features, such as learning, generalizing, fast real-time computation and modeling and classification capabilities.[51] For example, MacNamara *et al.*[52] used ANN in combination with Roman letters in the identification of the language of the entries in a library catalogue[51]; applied ANN with frequency analysis of letters in the identification of languages in multilingual documents; Selamat and Ng[41] also implemented ANN with letter frequency on web page language identification.

Figure 4 shows the example of architecture of Backpropagation Neural Network (BPNN) identification.[40,41] This BPNN consists of one input layer (p), one hidden layer (q) and one output layer (r). The total nodes of input layer depend upon the feature size, s, used for capturing input patterns. If the feature size, s, is 15 then the number of input layers will be set correspondingly. The number of one hidden layer

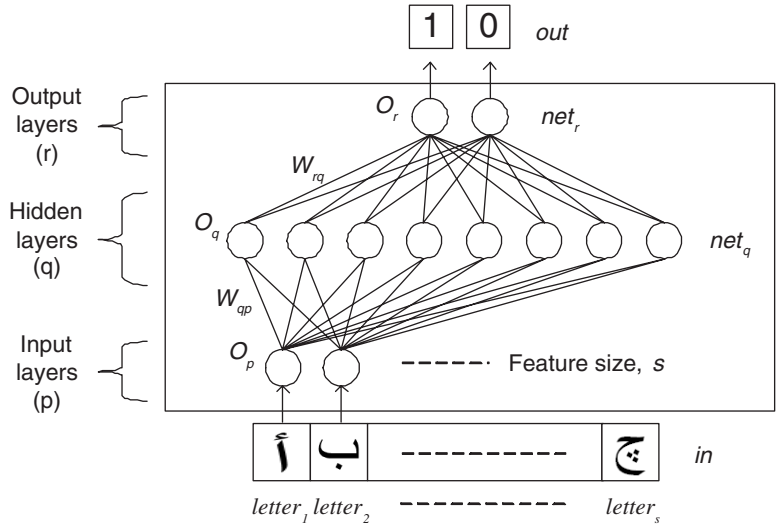

Fig. 4. Backpropagation neural networks architecture.

Table 1. Orthogonal language codes.

Language	Corresponding Vector
Arabic	0;0
Persian	0;1
Urdu	1;0
Jawi	1;1

is eight units. The number of an output layer consists 2 units, which is based on the corresponding output. Table 1 shows the corresponding orthogonal language codes used in the backpropagation neural network output layer. The output are binary forms that (0 0) represent Arabic language, (0 1) represent Persian language, (1 0) represent Urdu language, and (1 1) represent Jawi language, respectively. However, at times the orthogonal language codes sometimes might be different due to the design of BPNN architecture. Usually, the actual output produced by the model is compared with the desired output in order to insure the accuracy of the model. It is a justification of the model performance.

The neural networks parameters are defined as ℓ for the iteration number, t for the number of letter in a document, η for the learning rate, Γ for the momentum rate, Op for the output on unit p, Oq for the output on unit q, Or for the output on unit r, Wqp for the q^{th} weight to the unit p^{th}, Wrq for the r^{th} weight to the unit q^{th}, net_q is for the first transfer function at hidden layer q, net_r for the second transfer function at output layer r, θ_q is for the bias on hidden unit q, θ_r is for the bias on output unit r, δ_q is for the generalized error through a layer q, and δ_r is for the generalized error through a layer q and r. The input values of the backpropagation

neural network are represented by *in* where *in* is between 0 and 1 ($in \in [0,1]$), where s is the number of features that have been selected. The output values to the backpropagation neural network are represented by *out* where $out \in [0,1]$ which are corresponding to the Table 1. Adaption of the weight between hidden (q) and input (p) layers is given by

$$W_{qp}(\ell+1) = W_{qp}(\ell) + \Delta W_{qp}(\ell+1), \tag{5}$$

where

$$\Delta W_{qp}(\ell+1) = \eta\delta_q O_p + \Gamma\Delta W_{qp}(\ell) \tag{6}$$

and

$$\delta_q = O_q(1-O_q)\sum_r \delta_r W_{rq}. \tag{7}$$

Note that the first transfer function at the hidden layer (q) is given by,

$$net_q = \sum_q W_{qp}O_p + \theta_q \tag{8}$$

and

$$O_q = f(net_q) = 1/(1+e^{-net_q}). \tag{9}$$

Adaptation of the weights between output (r) and hidden (q) layers is given by,

$$W_{rq}(\ell+1) = W_{rq}(\ell) + \Delta W_{rq}(\ell+1), \tag{10}$$

where

$$\Delta W_{rq}(\ell+1) = \eta\delta_r O_q + \Gamma\Delta W_{rq}(\ell) \tag{11}$$

and

$$\delta_r = O_r(1-O_r)(\ell_r - O_r). \tag{12}$$

Then the output function at the output layer (r) is given by,

$$net_r = \sum_r W_{rq}O_q + \theta_r \tag{13}$$

and

$$O_r = f(net_r) = 1/(1+e^{-net_r}). \tag{14}$$

Table 2 shows the parameter setting on the BPNN. The input node, hidden node and output node are 42, 21 and 2, respectively; the learning rate is 0.001, the momentum rate is 0.0001, the epochs are 1,000, the minimum RMSE is 0.01, the features are normalized between -1 and 1 and output are normalized between 0 and 1.

Table 2. The BPNN structure.

Description	Value
Function	Logistic
Input Node	42
Hidden Node	21
Output Node	2
Learning Rate	0.001
Momentum Rate	0.0001
Epochs	1000
RMSE	0.01
Features Normalized	−1 to 1
Output Normalized	0 to 1

3.4. *KNN-SVM*

As the noisy training data in SVM must be discarded before a learning process can be done in the SVM Classifier, the k-nearest neighbor (KNN) method has been used to edit the training data set before it is used as an input to the SVM learning. The KNN training process will split all examples of data sets S into n classes. In this experiment n is refers to four languages of Arabic text, which are Jawi, Persian, Urdu and Arabic. The average number of data sets in each set of classes has been constructed in order to classify all the training data sets. The misclassified data sets must be discarded by using the KNN training $f(i+1)$. Finally, the KNN algorithm will be used to classify the remaining examples in order to build the SVM classifier.

There are two parts to KNN-SVM classifier as shown in Fig. 5. They are the training process and the testing/prediction process. The training process consists of the concurrent hybrid KNN and SVM approaches. The KNN-SVM algorithm consists of two steps for the training process. These are the KNN-SVM training and the SVM training. The KNN-SVM training will return the SVM model as the result and then the SVM classifier will use the model to predict the language of the Arabic script document under test.

KNN-SVM training steps:

Step 1. Classify all sample data sets using KNN algorithm by finding the k nearest distance from its document using an Euclidean formula $\sqrt{\sum_{i=1}^{n}(x_{A,i} - x_{B,i})^2}$.

Step 2. Remove all misclassification sample data sets from the training set.

$$\text{If } \mathit{KNNClassifier}(\mathit{Doc}[i]) \neq \mathit{ClassOf}(\mathit{Doc}[i]) \quad \text{then } \mathit{remove}(\mathit{Doc}[i])$$

Step 3. Repeat Step 1 until the misclassification of data is not found

Step 4. Using a clean sample of data set in order to find the hyperplane and support vectors.

$$\mathit{minimize}\ \frac{1}{2}\|w\|^2, \quad \mathit{Subject\ to}\ y_i(w^T + b) \geq 1$$

Step 5. Save the SVM model as the result of classifier.

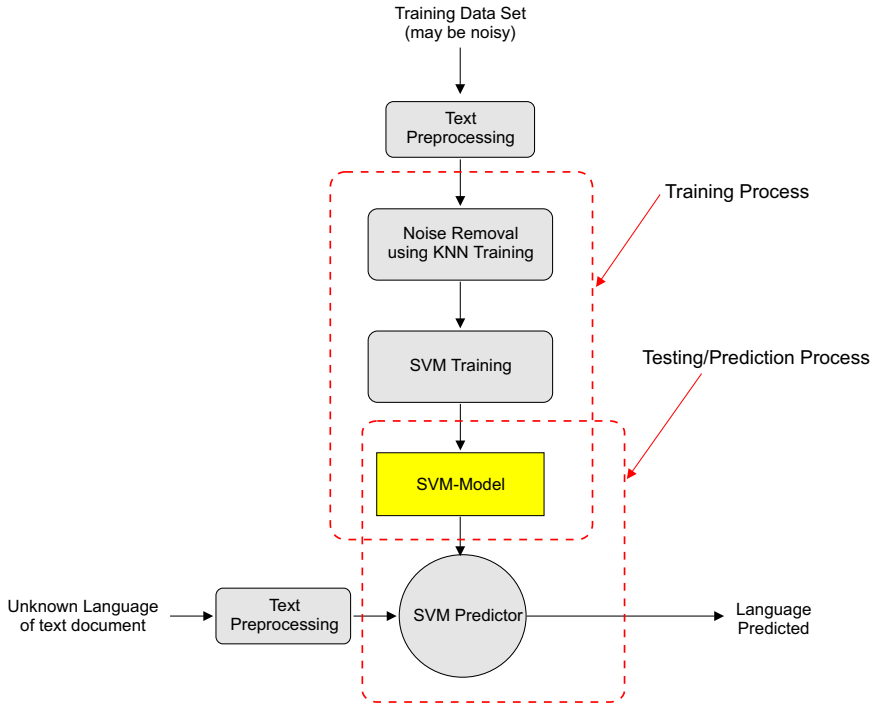

Fig. 5. KNN-SVM training and prediction process.

The text preprocessing converts the original data (web page document) to vector space model (VSM) data. It involves data cleaning and character frequency calculation. In the training process in Fig. 5 there are two processes needed to acquire the SVM-model, they are KNN training and SVM training. The KNN training is used for reducing the data set which is detected as misclassified data. Its data will be reduced because it affects the accuracy of the classifier. The next step is SVM training, where training data is produced by KNN and learnt by SVM model. The output of the training part is the SVM model that will be used by the classifier to classify the unknown language web document that uses Arabic script. The reason of using the KNN before the SVM training is to protect the SVM classifier from the insensitivity of misclassification data training. This is explained in Fig. 6.

Figure 6 describes the algorithms of the KNN-SVM training process by a two dimensional figure. Figure 6(a) shows the difficulty of the SVM training in finding the hyperplane separator, especially for a linear regression. Figure 6(b) shows the noise removal process using the KNN classifier, as the misclassified data set will be removed. The expectation of this process is that it will ease the finding of the hyperplane in SVM training. Figure 6(c) shows the SVM training function that will more easily find the hyperplane after noise removal process.

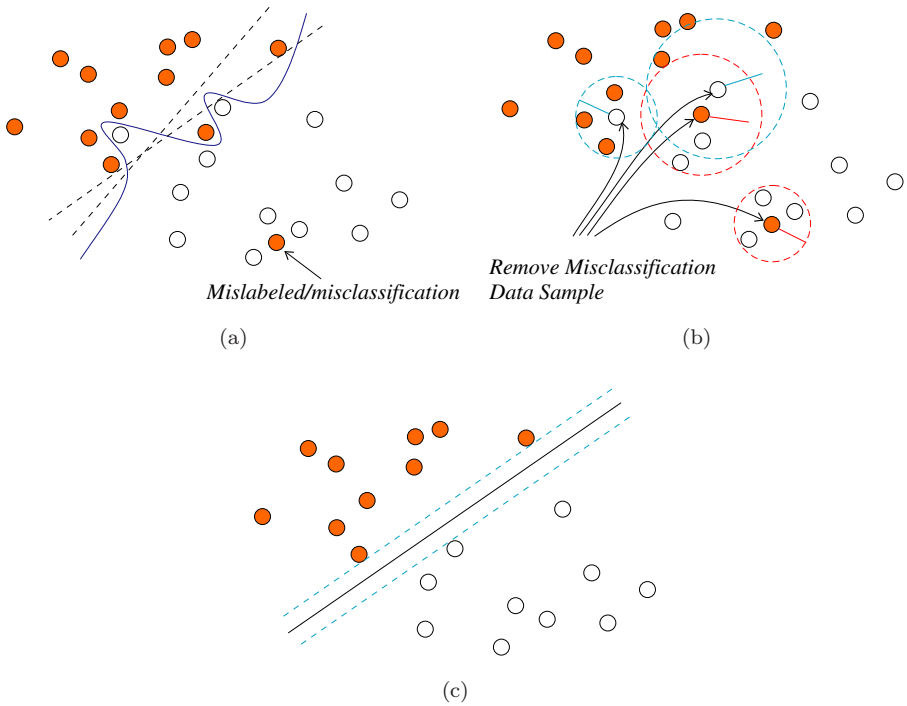

Fig. 6. (a) The difficulty of the SVM hyperplane in finding the correct data for classification, (b) The process of cleaning the training data sets using KNN, (c) The SVM training using clean data set.

3.5. *KNN-BPNN*

Similar to the step involving on KNN-SVM, the KNN has been used to filter the data and then fed into BPNN for training. It is also divided into training and testing processes. The training process involves both KNN and BPNN, but the testing data has been feed directly into trained BPNN for prediction. Figure 7 illustrates the idea of KNN-BPNN. The only difference is in Steps 4 and 5, where the trained data of KNN is fed into BPNN to be trained again. Therefore, the KNN-BPNN step is derived in the following manner:

Step 1. Classify all sample data sets using KNN algorithm by finding the k nearest distance from its document using an Euclidean formula $\sqrt{\sum_{i=1}^{n}(x_{A,i} - x_{B,i})^2}$.

Step 2. Remove all misclassified sample data sets from the training set.

$$\text{If } KNNClassifier(Doc[i]) \neq ClassOf(Doc[i]) \quad \text{then } remove(Doc[i])$$

Step 3. Repeat Step 1 until a misclassification of data is not found.

Step 4. Iterate the training process of BPNN until the error convergence is achieved (Repeat from Eqs. (5) to (14))

Step 5. Save the weight of the BPNN for prediction.

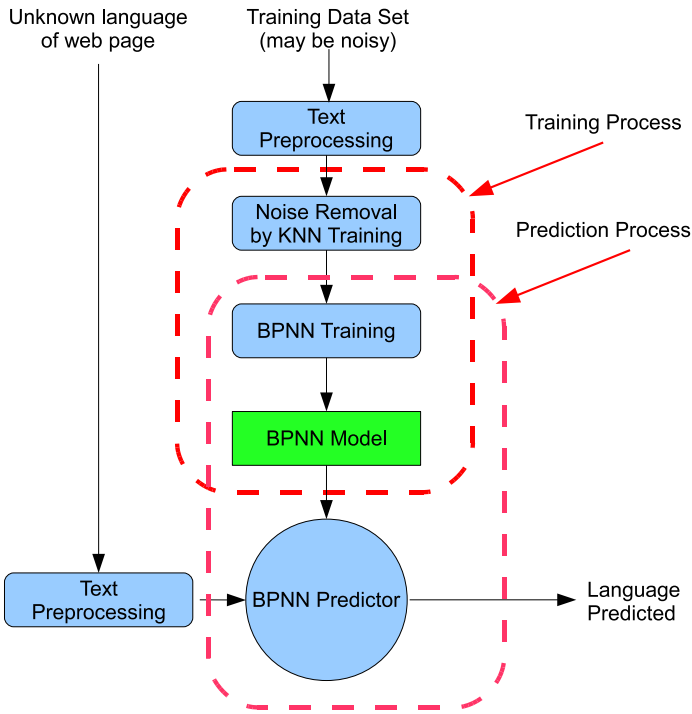

Fig. 7. KNN-BPNN training and prediction process.

4. Preprocessing and Evaluation Measurement

Preprocessing involves document collection, document representation, preprocessing and feature selection is involved. The evaluation measurements that have been utilized, such as precision, recall and F1 measurements, are discussed in the following section.

4.1. *Document collection*

We have acquired the news data set from the British Broadcasting Corporation (BBC) website.[53] Those data sets (2–10 kb) saved in Unicode form by setting the file name corresponding to their languages. For instance, text collected from selected Arabic BBC news is saved as "*a1.txt*" (a1 means first Arabic document). This process is repeated for 200 documents that were collected for each language. We organized the web page collection by manually assigning language tags to one of four different languages (Arabic, Persian, Urdu and Pashto). In this way the tagged Unicode documents were prepared for evaluation (Fig. 8).

4.2. *Document representation*

Web page language identification is defined as the task of assigning a collection of web documents $D = d_1, d_2, \ldots, d_{|\mathcal{D}|}$ to a set of predefined categories of Arabic script

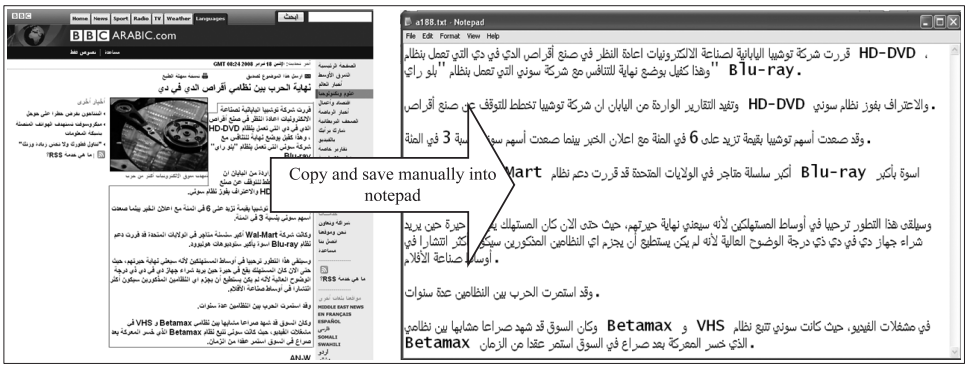

Fig. 8. Example of way collecting document from BBC website.

languages $L = l_1, l_2, \ldots, l_{|L|}$. Many web page language identification methods are based on vector space model (VSM)[48] representations. As a result, each document (d_j) is defined as a vector of weights (w_j) related to the language terms corresponding to text. Thus, a document corpus containing $|\mathcal{D}|$ documents and $|\tau|$ language terms is represented by means of a term in a document matrix $\mathbf{X}$ as follows:

$$[h]\mathbf{X} = \left(\begin{array}{c|ccccc} & d_1 & d_2 & d_3 & \ldots & d_{|\mathcal{D}|} \\ \hline t_1 & w_{11} & w_{21} & w_{31} & \ldots & w_{|D|1} \\ t_2 & w_{12} & w_{22} & \ldots & \ldots & \vdots \\ t_3 & \vdots & \ddots & \ddots & \ldots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ldots & \vdots \\ t_{|\tau|} & w_{1|\tau|} & \ldots & \ldots & \ldots & w_{|\mathcal{D}||\tau|} \end{array}\right). \tag{15}$$

However, we are using matrix $\mathbf{Y}$ that has a similar approach to matrix $\mathbf{X}$. Matrix $\mathbf{Y}$ is based on the character frequency (CF) of the documents, while weight is the frequency of character in the document. In contrast, matrix $\mathbf{X}$ uses TFIDF as the weight based on the term frequency; usually a term is a word. When using the term based method, stemming and stopping is very important to improve the retrieval performance but with the character frequency based method, the stemming and stopping are not necessary. The important preprocessing for this approach is the cleaning of the web page documents to the plain text. As a note, the encoding system conversion is also important for web documents.

The weight is actually the expression of how important a certain feature that represents a documents. We have assumed that character frequency is the important aspect of document representation. In our hypothesis each language in the Arabic script has its own pattern of character frequency.

Based on the VSM model, each document d can be interpreted as a vector in a character space. In the simplest form, each document can be represented via a

character frequency vector as follows:

$$[h]\mathbf{Y} = \left(\begin{array}{c|ccccc} & d_1 & d_2 & d_3 & \dots & d_{|\mathcal{D}|} \\ \hline c_1 & cf_{11} & cf_{21} & cf_{31} & \dots & cf_{|D|1} \\ c_2 & cf_{12} & cf_{22} & \dots & \dots & \vdots \\ c_3 & \vdots & \ddots & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \dots & \vdots \\ c_{|\tau|} & cf_{1|\tau|} & \dots & \dots & \dots & cf_{|\mathcal{D}||\tau|} \end{array}\right). \tag{16}$$

$$d_{cf} = (c_{f_1}, c_{f_2}, c_{f_3}, \dots, c_{f_n}). \tag{17}$$

where c_{fi} is the frequency of the ith character in the document. As a weighting model for the CF vector, the frequency-weighting vector is chosen. Therefore, for each document CF vector will also be its weighting vector. At the last step, normalization is achieved by transforming each document vector into a unit vector. In this model, documents can be imagined as points in a character space and therefore the similarities among documents can be calculated by geometrical methods. The common distance measurement is the Euclidean distance formula.

The size of the vector affects the execution time for training or testing processes and also affects their accuracy. A large quantity of data makes the process slower while a smaller quantity of data might cause a decrease in accuracy. The feature selection method is used to reduce the number of features that have a significant impact on the classifier. Sometimes the heuristics method is suitable. The method for feature selection that was used in this experiment will be explained later in this paper.

In language identification studies, one of the main problems is the dimension of the feature set. Generally, feature sets are constructed from n-grams or short terms and these are very large in size. Therefore, reducing their dimension is necessary in language identification studies. Using characters in the language identification process will in most intones solve the dimension problem. For example, the numbers of n-grams and common words are estimated as 2,550–3,560 and 980–2,750, respectively, as stated by Grefenstette.[54] However, there are 25–50 characters on average for alphabets and therefore, using characters as a feature set will have more advantages. Figure 9 shows the processing of the web documents for identification of the text document languages from the Internet based on character frequency (CF).

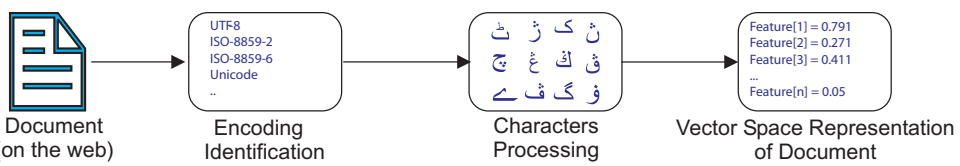

Fig. 9. The pre-processing of web documents for an Arabic script language identification process based on character frequency (CF).

4.3. *Preprocessing*

Many encoding types in web documents need to be identified in order to ensure the processing of the characters will not be miscalculated. The web document encoding can be identified by the header part of the HTML document which contain a "charset = (encoding type)", for example "charset = utf-8". This encoding type is very important for character processing (see Fig. 9) in order to ensure the correct display of the characters on the browsers that correspond to the language of the text. Converting the identified encodings to Unicode encoding is useful because the Unicode encoding will be able to accommodate all encoding types of characters that appear in a web document into a specific numeric number. The document must be cleaned from HTML tags before it can be transformed the texts into character frequency. In character-based method, the cleaning, stemming and stopping process of the web document is not necessary. Documents need to be normalized according to their length as shown in Eq. (18). From 1,200 samples of web documents in Persian, Jawi, Urdu and Arabic languages, we have normalized the character frequency by the maximum frequency among those document using Eq. (19) as follows:

$$X_i = \frac{f_{c_i}}{NC}, \tag{18}$$

$$X_i' = \frac{X_i}{X_{\max}}, \tag{19}$$

where NC represents the number of character in a document, f_{c_i} represents frequency of ith character, c represents character, i represents character number in document i. After normalization the value of X_i' will be at intervals of 0 to 1.

4.4. *Feature selection*

The next step after the pre-processing of documents is the feature selection and representation processes. We have used the feature selection method to remove some irrelevant or inappropriate features from the feature set. The aim of this process is to find the best features that will be able to represent most of the documents content. The purpose is to reduce the computational space and time. In character-based language identification, characters are chosen as features. We have chosen only the alphabet of the respective languages in order to reduce dimensional problems. Based on the alphabet of the four languages, we have randomly selected 80 documents and calculated the frequency of each character and then calculated the root mean-square error (RMSE) of each of the characters frequency from its average. The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}}, \tag{20}$$

where n is number of documents, x_i is the frequency of ith document and $\overline{x}$ is the mean of character frequency for all the documents. Only the frequency with

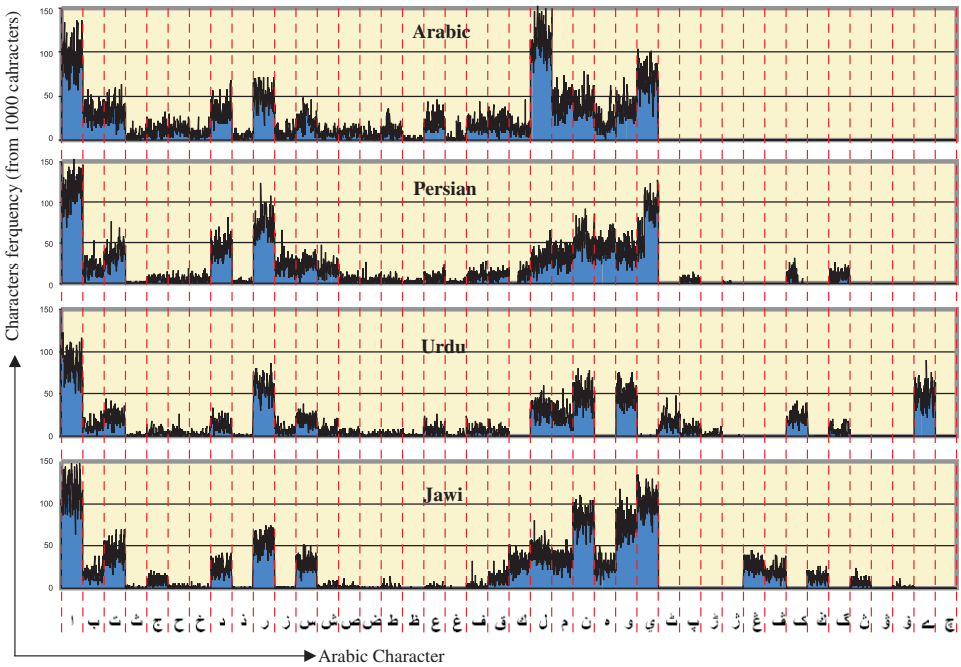

Fig. 10. Arabic pattern based on characters frequency.

a small error will be chosen as the feature set. Figure 10 shows the pattern for each language based on character frequency. We have only selected the first 1,000 characters from each document as a standard for our data set.

Figure 11 shows the pattern of Arabic language based on frequency average referred in Fig. 10. Each of the character's frequency in Fig. 11 has been computed based on its average and then from the average we have found the error deviation. Average and standard deviation using RMSE; the centroid based classifier (CB) can also be used. If the deviation is smaller, it indicates that the feature is suitable for the CB classifier. Usually, if a case can be solved by the CB classifier, other classifiers such as KNN and SVM can be applied.

Arabic feature sets are (from ا to ي) chosen from Arabic alphabet. The alphabet of each language is chosen because some other characters like ?, #, &, %, etc., are not counted as the features in our research. All the four languages (Arabic, Persian, Urdu and Jawi) use these characters, but the three last languages have some additional unique characters. Finally we defined 42 unique characters alphabetic that accommodate all four languages as shown in Table 3. The usage of 42 character as a feature set reduces the volume of a feature set dramatically. Before we select the 42 features, all alphabets of the four languages (Arabic, Urdu, Persian, and Jawi) had been calculated based on their frequency and the mean squared error (MSE) for each character in order to measure the consistency of frequency of each

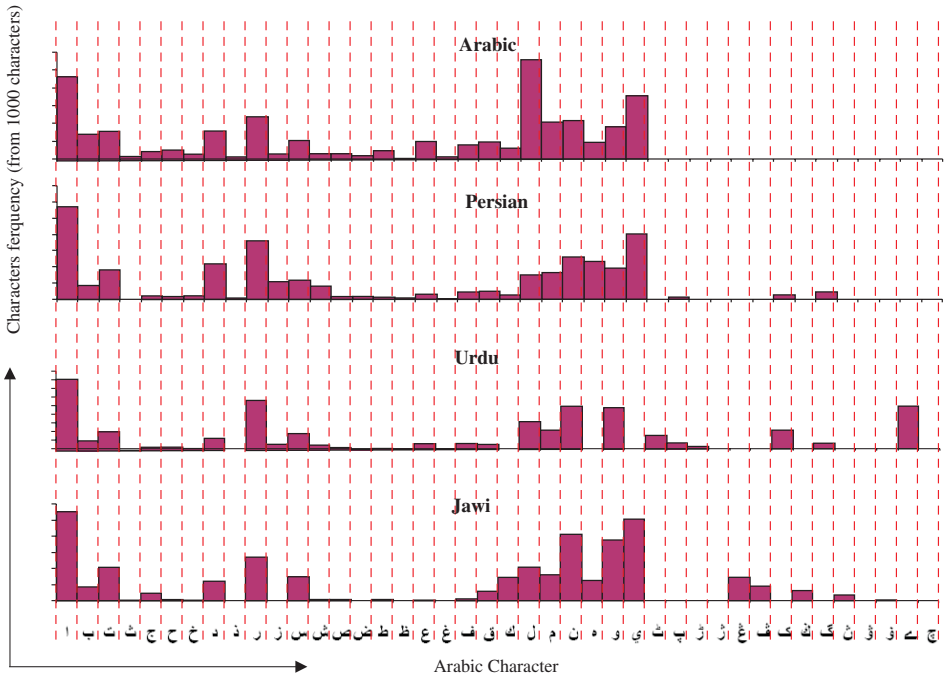

Fig. 11. Arabic pattern based on characters frequency average.

Table 3. The unique characters of four Arabic script languages.

Feature ID	Character	Unicode Number	Arabic	Persian	Urdu	Jawi
1	ا	#1575	√	√	√	√
2	ب	#1576	√	√	√	√
3	ت	#1578	√	√	√	√
⋮	⋮		⋮	⋮	⋮	⋮
28	ي	#1610	√	√	√	√
29	ڽ	#1657			√	
30	ڤ	#1662		√	√	
31	ٹ	#1681			√	
32	ټ	#1688		√	√	
33	ڠ	#1696				√
34	ڤ	#1700				√
35	ک	#1705			√	
36	ڬ	#1708				√
37	گ	#1711		√	√	
38	ڻ	#1725				√
39	ۏ	#1739			√	
40	ۋ	#1743				√
41	ۿ	#1746			√	
42	ﭽ	#64378		√		√

document. From 1,292 documents in the experiment, the average of MSE is 0.00212. This means that the consistency of characters frequency in many documents is very high. Therefore, all the 42 features available from the data set have been chosen as the final feature set.

4.5. *Evaluation measurements*

The proposed methods are evaluated using the standard of information retrieval measurements that are precision (p), recall (r), and $F1$. They are defined as follows:

$$\text{precision} = \frac{a}{a+b}, \tag{21}$$

$$\text{recall} = \frac{a}{a+c}, \tag{22}$$

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}, \tag{23}$$

where the values of a, b and c are defined in Table 4. The relationship between the classifier and the expert adjustment is expressed using the four values as shown in Table 5.

The precision describes the probability that a retrieved Arabic document (randomly selected) retrieved document is relevant to the certain language. The recall indicates the probability that a relevant Arabic document is retrieved. The overall evaluation measure of this LID on Arabic script is $F1$ which describes the average between precision and recall.

The noise tolerance performance of SVM training has been measured by the percentage of the noises that are removed by KNN training. Ideally all noises will be removed before the SVM training process, so that the SVM-training is able to

Table 4. The definitions of the parameters a, b and c which are used in Table 5.

Value	Meaning
a	The system and the expert agree with the assigned category
b	The system disagrees with the assigned category but the expert did
c	The expert disagrees with the assigned category but the system did
d	The system and the expert disagree with the assigned category

Table 5. The decision matrix for calculating the classification accuracies.

Expert	System	
	Yes	No
Yes	a	b
No	c	d

use the clean training data set. If the training process uses clean data then the testing performs better.

5. Experimental Results

We have conducted three experiments in order to evaluate the performance of the proposed methods on Arabic script web page language identification. The first experiment is the identification performance comparison using confusion matrix. The second is to justify the retrieval performance using the perspective of precision, recall and $F1$ measurement. The last experiment observes the noises removal performance by the KNN method.

5.1. *Experiment 1: Identification performance comparison*

The objective of experiment 1 is to test the language identification performance of each method KNN, SVM, BPNN, SVM-KNN and BPNN-KNN. This is to review the ability of a particular method in determining the language of the raw data after preprocessing and we find that BPNN and KNN-BPNN are superior to others. In this experiment, the 1,100 samples are normalized using Eqs. (18) and (19). Then, the normalized samples are divided between 400 samples for training and 700 for testing. The results of this experiment is shown in Tables 6–10, respectively, in the form of a confusion matrix that measures the desired language against predicted language.

Table 6 shows the result of web page language identification using the KNN classifier. It is noticed that the predicted language of Persian data is one Arabic,

Table 6. KNN classifier testing accuracy.

Original Language	Prediction				Accuracy (%)
	Arabic	Persian	Urdu	Jawi	
Arabic	200	0	0	0	100.00
Persian	1	190	0	9	95.00
Urdu	0	0	200	0	100.00
Jawi	0	0	0	100	100.00
Average					98.75

Table 7. SVM classifier testing accuracy.

Original Language	Prediction				Accuracy (%)
	Arabic	Persian	Urdu	Jawi	
Arabic	197	2	0	0	98.99
Persian	0	200	0	0	100.00
Urdu	0	1	199	0	99.50
Jawi	0	0	5	95	95.00
Average					98.37

Table 8. BPNN classifier testing accuracy.

Original Language	Prediction				Accuracy (%)
	Arabic	Persian	Urdu	Jawi	
Arabic	200	0	0	0	100.00
Persian	0	200	0	0	100.00
Urdu	0	0	200	0	100.00
Jawi	0	0	0	100	100.00
Average					100.00

Table 9. KNN-SVM classifier testing accuracy.

Original Language	Prediction				Accuracy (%)
	Arabic	Persian	Urdu	Jawi	
Arabic	197	2	0	0	98.99
Persian	0	200	0	0	100.00
Urdu	0	1	199	0	99.50
Jawi	0	0	5	95	95.00
Average					98.37

Table 10. KNN-BPNN classifier testing accuracy.

Original Language	Prediction				Accuracy (%)
	Arabic	Persian	Urdu	Jawi	
Arabic	200	0	0	0	100.00
Persian	0	200	0	0	100.00
Urdu	0	0	200	0	100.00
Jawi	0	0	0	100	100.00
Average					100.00

190 Persian and 9 Jawi, respectively, and the accuracy of KNN classifier is 95%. Overall, the average accuracy of KNN classifier is 98.75%.

Table 7 shows the confusion matrix of the SVM classifier accuracy in web page language identification. The output of Arabic data is 197 Arabic, 2 Persian and one datum is misclassified from existing language. For the Urdu data, one was predicted as Persian and others were correctly predicted as Urdu. Moreover, the output of Jawi data is five data was identified as Urdu and the rest as Jawi. The accuracy of Arabic, Persian, Urdu and Jawi is 98.99%, 100%, 99.50% and 95.00%, respectively. The average accuracy of SVM classifier is 98.37%.

Table 8 shows the testing accuracy using the BPNN classifier. Overall, the BPNN classifier is able to correctly determine all the desired languages. Although SVM is the most recommended for actual application of classification, the results show that BPNN is better than the KNN and SVM classifiers in web page language identification. With the KNN classifier, the document to be classified has to calculate the distance from all other classified samples. Therefore, the KNN performance

very dependent on the size of samples; the smaller the size of sample, the greater the risk of decreasing accuracy, but it may perform faster. For example, if 400 samples are used then the KNN classifier must calculate the distance from the document to other sample documents, or make the equivalent of 400 calculations. In SVM, the data is classified according to the coordinates of the document based on the support vectors, so that the SVM can perform very fast. For BPNN, the process training is time consuming due to the need to minimize the error rate by iterating the training process. BPNN performs robustly in the prediction if the training data given achieve fast error convergence, or the training data is for the most part free from noise. It can be concluded that BPNN is most recommended in this case.

Table 9 shows the hybrid method between KNN and SVM or KNN-SVM in the web page language identification. We have observed that the output of KNN-SVM is the same as the output of SVM, as shown in Table 7. The same situation occurred with another hybrid method between KNN and BPNN or KNN-BPNN, as shown in Table 10, the output of the prediction is same as that shown in Table 8. At this stage, we cannot conclude that the hybrid-KNN method is not an improvement over the conventional method because the clean training sets used in this part of the experiment. If the training data set is clean, then the output of the KNN training is almost the same because there are no misclassification of data or noise since they have been removed. In the hybrid-KNN, the output of the KNN training data set is then used as the training data of particular hybrid method. Therefore, the training data used either for KNN-SVM or KNN-BPNN is almost same as the training data used on SVM and BPNN, respectively. It is easily concluded that the output of the hybrid method or orginal method will be the same if clean training data was used. The improvement of hybrid-KNN method can be analyzed from the data training that has some degree of misclassification data or so-called noise, which will be presented in the following section.

5.2. *Experiment 2: Retrieval performance*

The objective of experiment 2 is to measure the retrieval performance of the proposed methods on Arabic script web page language identification. The measurements used in this experiment are precision, recall and $F1$ measurements. The precision describes the probability that a randomly selected and retrieved Arabic document is relevant to the certain language. The recall describes the probability of a relevant Arabic document being retrieved. The $F1$ measure is the average between precision and recall. Table 11 shows the retrieval performance of KNN, SVM, BPNN, KNN-SVM and KNN-BPNN, respectively. From the 400 documents used for training and 700 documents used for testing, all the results show a high level of accuracy is above 95%. It can be concluded that the choice of document representation using character frequency is suitable for Arabic script web page language identification. Furthermore, both BPNN and KNN-BPNN are shown to be superior than others in precision, recall and $F1$ measurements, with the accuracy 100%. This

Table 11. The retrieval performance of Arabic script web page language identification in terms of precision, recall and $F1$ measurement.

Languge Class	Arabic	Persian	Urdu	Jawi	Average (%)
KNN					
Precision (%)	100.00	95.00	100.00	100.00	98.75
Recall (%)	99.50	100.00	100.00	91.74	97.81
$F1$ (%)	99.75	97.44	100.00	95.69	98.22
SVM					
Precision (%)	98.99	100.00	99.50	95.00	98.37
Recall (%)	100.00	98.52	97.55	100.00	99.02
$F1$ (%)	99.49	99.26	98.51	97.44	98.68
BPNN					
Precision (%)	100.00	100.00	100.00	100.00	100.00
Recall (%)	100.00	100.00	100.00	100.00	100.00
$F1$ (%)	100.00	100.00	100.00	100.00	100.00
KNN-SVM					
Precision (%)	98.99	100.00	99.50	95.00	98.37
Recall (%)	100.00	98.52	97.55	100.00	99.02
$F1$ (%)	99.49	99.26	98.51	97.44	98.68
KNN-BPNN					
Precision (%)	100.00	100.00	100.00	100.00	100.00
Recall (%)	100.00	100.00	100.00	100.00	100.00
$F1$ (%)	100.00	100.00	100.00	100.00	100.00

corroborates the explanation that the data set which is used for this experiment is a clean data set.

5.3. *Experiment 3: Noises removal performance*

Table 12 shows the result of the retrieval performance against the level of noise added in the Arabic script web page. The level of noise used is 0%, 2%, 4%, 8%, 15%, 20%, 25%, 30%, 35%, 40%, 45% and 50%. There are 400 samples used for training, so if the level of noise is 2%, then eight samples of 400 data will be changed to the wrong desired language, and so on. The average result of KNN, SVM, BPNN, KNN-SVM and KNN-BPNN is 81.83%, 90.79%, 99.91%, 96.78% and 93.69%, respectively. It is noticed that KNN and SVM cannot maintain the accuracy of identification when there is an increase in the level of noise. There is a significant drop-off found in KNN-BPNN, which the accuracy of identification is 25% only when the level of noise is 15%. However, the accuracy of BPNN is 100% and KNN-SVM also has a decrement to 96.98% at same level. Therefore, it is assumed that the data filter by KNN has the potential to increase the noise in the data which might not be suitable for BPNN. Although KNN is well known for clustering and data filtering, it seems as if it is not suitable in the case of Arabic script web page language identification. We have observed that the BPNN is superior to others against the level of noise added in data. Moreover, KNN-SVM performs more reliably than KNN-BPNN against the added noise.

Table 12. Testing accuracy based on the level of noise in the data set of Arabic script web page.

Noise (%)	KNN	SVM	BPNN	KNN-SVM	KNN-BPNN
0	98.75	98.37	100.00	98.37	100.00
2	98.75	98.37	100.00	98.37	100.00
4	98.00	98.25	100.00	98.37	100.00
8	95.13	98.12	100.00	98.25	100.00
15	92.75	96.12	100.00	96.98	**25.00**
20	89.13	95.75	100.00	97.62	100.00
25	79.88	96.50	100.00	97.50	99.88
30	72.00	95.63	99.88	97.12	99.88
35	69.75	94.63	100.00	96.99	100.00
40	66.13	95.74	99.88	95.74	99.88
45	62.50	72.25	99.75	94.50	99.88
50	59.13	49.75	99.38	91.50	99.75
Average	81.83	90.79	99.91	96.78	93.69

The objective of experiment 3 is to test the noise tolerance performance of KNN against the noises inside the training data set. In this experiment, we have been able to prove that KNN-SVM improves the accuracy and immunity of SVM from the training data noises. The hypothesis is that an improvement can be made by KNN as a part of the pre-training of SVM. Figure 12 shows the result of the noise removal using KNN and the number of k is three. We have noticed that the KNN and SVM cannot be used independently against the level of added noise. The accuracy of KNN and SVM on the level of 50% noise is 59.13% and 49.75%, respectively. However, the hybrid method KNN-SVM is able to maintain the accuracy of identification above 91.5%. Although the KNN-BPNN has a problem with noise tolerance, this experiment shows that the hybrid method KNN-SVM performs reliably against noise in the Arabic script web page language identification against noise.

Figure 13 shows that KNN is effective in removing the noisy training data. The noise removal performance was measured by calculating the percentage of the noise

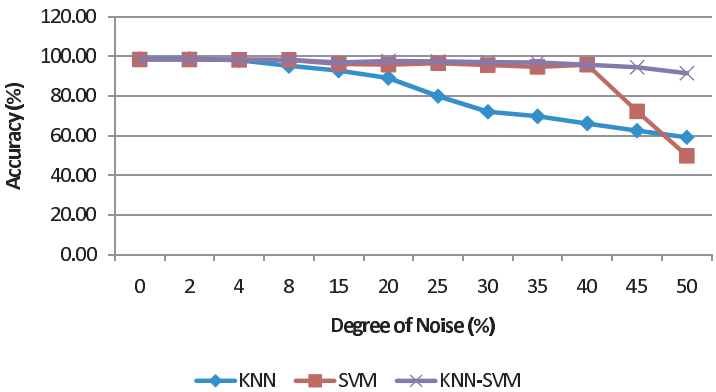

Fig. 12. The impact of KNN on the noise tolerance of Arabic script language identification.

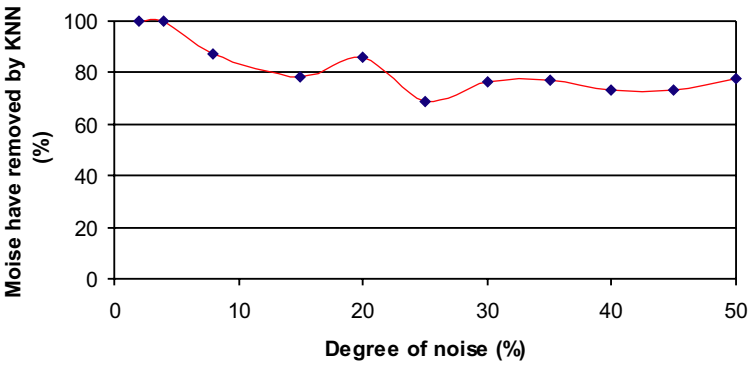

Fig. 13. Noise removal performance by KNN.

which was removed by KNN training. From the figures, we can observe that the KNN-SVM is able to give high performance even when the noise degree reached 50% of the training data set. The average of all testing of noises removal performance was 81.7%. This means that KNN with $k = 3$ is effective in removing the noise.

6. Discussion

In this work, we have researched the problem of Arabic script web page language identification by proposing hybrid-KNN methods. In the first experiment, five methods are compared for the identification performance. The conventional methods are KNN, SVM and BPNN. This is extended to another two hybrid methods, namely KNN-SVM and KNN-BPNN. The KNN acts as a noise filtering method. Overall, in the identification performance on raw data, the BPNN and KNN-BPNN are superior to others, since both methods are able to identify completely the particular web page desired language. However, KNN does not impact the identification performance due to the fact that the data used was clean.

In the second experiment, another analysis on the retrieval performance of Arabic script web page language identification was carried out. For the retrieval performance, both BPNN and KNN-BPNN remained the best identifier, with their precision, recall and $F1$ measurements achieving the highest percentages. The BPNN is robust in prediction if the data set given on training is sufficiently clean. However, the question is how reliable is the BPNN against added noise or misclassification data found. Can it still perform stably?

The third experiment is designed to evaluate the noise tolerance performance by KNN. We have observed that the BPNN performs best with respect to noise tolerance performance. However, the hybrid method KNN-BPNN shows a significant drop-off when the level of noise is 15%. Therefore, it is assumed that KNN-BPNN is not reliable when the noise is naturally found in the data or produced by the KNN. In the actual application, the data used for training and testing is supposed to be

extensive. Moreover, the number of languages should be more than the four languages discussed here. As a consideration, the web statistics show that the number of languages in the world is 6,912.[7] Therefore the data collection for training and testing through the use of automation will be very diffcult. The alternative possible method is manual collection. This is a basic problem since manual collection has a high risk of data misclassification. Therefore, we have proposed the hybrid KNN-SVM method as the solution to filter the misclassified data for training and testing. KNN-SVM is a good instrument as shown in Fig. 12. The KNN training will remove almost all misclassification before SVM training. The SVM is a better classifier than KNN (see result in Fig. 12) and is faster than the KNN classifier. As a result, it is proven that KNN-SVM performs most reliably against the noise or misclassification data found.

The high identification performance was affected strongly by feature representation and selection. In this work, character frequency was used as the feature that represented it as the vector space model (VSM). Figure 10 shows that the pattern is strong and centroid based (CB). This was proven by the small value of RMSE, which can be interpreted as that when the error deviation is small and it will have no impact. It can be concluded that this feature representation is strong. The feature representation that is suitable for CB would be suitable too for other techniques of classifiers such as KNN, SVM and BPNN (as shown in Table 12).

In future works, the data set used can be extended into others script data such as Latin, Hanzi, Cyrillic, Indic, etc. The results of the identification may vary as the nature of each language, for example the varying morphological composition of languages. This will affect the letter distribution, which can then directly have an impact on the identification performance. Moreover, the proposed hybrid-KNN can also be applied to the multilingual document. The multilingual document may consist of 70% Arabic and 30% Indonesia, or both mixed languages of Arabic and Urdu. This problem is more complex than the mono-lingual identification issue due to the fact that the boundary of a language and the similarity between languages is highly confusing.

7. Conclusion

In this work, we have presented Arabic script web page language identification using hybrid-KNN methods and based on character frequency distribution. We have selected 42 alphabets as features that can accommodate four languages to reduce the dimensional space of machine learning methods. The character frequency is proven as good features in Arabic script web page language identification, in which the identification performance can be maintained above 95%. The KNN has been selected as the filtering method in combination with other supervised machine learning methods on language identification. It has been proven that KNN with $k = 3$ can reach a noise tolerance performance at 81.7% on average. The results also have shown that KNN-SVM is an enhancement over conventional methods in

identifying the misclassification data instead of KNN-BPNN, even on the level of 50% noise. The average accuracy of noise tolerance performance of KNN-SVM and KNN-BPNN is 96.78% and 93.69%, respectively. This has proven that KNN-SVM is reliable on Arabic script web page language identification. In future, we plan to further analyze the applicability of the KNN-SVM approach to identify more languages in different scripts such as Thai, Tamil and Urdu.

Acknowledgement

This work is supported by the Ministry of Science, Technology & Innovation (MOSTI), Malaysia and Research Management Center, Universiti Teknologi Malaysia (UTM), under the Vot 79200 and 79267. The authors would like to thank Prof. Richard L. Spear for his valuable suggestions in improving the manuscript. The authors are also grateful to the anonymous reviewers for their valuable and insightful comments.

References

1. R. G. Gordon and B. F. Grimes, *Ethnologue: Languages of the World* (SIL International, USA, 2005).
2. M. Z. Abd Rozan, Y. Mikami, A. Z. Abu Bakar and O. Vikas, Multilingual ict education: Language observatory as a monitoring instrument, *Proc. South East Asia Regional Computer Confederation (SEARCC) 2005: ICT Building Bridges Conf.*, Vol. 46, Sydney, Australia (2005), pp. 53–61.
3. D. Maclean, Beyond English: Transnational corporations and the strategic management of language in a complex multilingual business environment, *Manag. Decis.* **44** (2006) 1377–1390.
4. I. Redondo-Bellon, The effects of bilingualism on the consumer: The case of Spain, *European J. Marketing* **33** (1999) 1136–1160.
5. T. Friedman, *The World is Flat* (Farrar, Straus and Giroux, New York, 2005).
6. M. M. Group, Internet world users by language: Top ten internet languages used in the web (2007), accessed 15 November 2007.
7. P. J. Payack, The global language monitor (2007), accessed 20 November 2007.
8. B. A. C. Comrie, Language: Microsoft encarta online encyclopedia (2007), accessed 10 December 2007.
9. J. D. Allen and C. Unicode, *The Unicode Standard 5.0* (Addison-Wesley, UK, 2007).
10. A. Selamat and I. Ibnu Subroto, Language identification of Arabic scripts web documents based on knn-svm, *Proc. 1st Int. Workshop on Hybrid Soft Computing in Engineering* (2007).
11. J. Capstick, A. Diagne, G. Erbach, H. Uszkoreit, A. Leisenberg and M. Leisenberg, A system for supporting cross-lingual information retrieval, *Info. Process. Manag.* **36** (2000) 275–289.
12. B. Mobasher, H. Dai, T. Luo and M. Nakagawa, Improving the effectiveness of collaborative filtering on anonymous web usage data, *Proc. IJCAI'01 Workshop on Intelligent Techniques for Web Personalization* (2001), pp. 1–8.
13. C. Yu, B. Ooi, K. Tan and H. Jagadish, Indexing the distance: An efficient method to knn processing, *Proc. 27th Int. Conf. Very Large Data Bases* (Morgan Kaufmann, San Francisco, CA, 2001), pp. 421–430.

14. T. Kudoh and Y. Matsumoto, Use of support vector learning for chunk identification, *Proc. 2nd Workshop on Learning Language in Logic, 4th Conf. Computational Natural Language Learning*, NJ (2000), pp. 142–144.
15. W. Campbell, E. Singer, P. Torres-Carrasquillo and D. Reynolds, Language recognition with support vector machines, *ODYSSEY04 — The Speaker and Language Recognition Workshop*, *ISCA'04* (2004).
16. H.-Z. Li, B. Ma and C.-H. Lee, A vector space modeling approach to spoken language identification, *IEEE Trans. Audio, Speech, and Language Processing* **15** (2007) 271–284.
17. J. Zou, G. Chen and W. Guo, Chinese web page classification using noise-tolerant support vector machines, *Proc. 2005 IEEE Int. Conf. Natural Language Processing and Knowledge Engineering* (2005), pp. 785–790.
18. P. Sibun and J. C. Reynar, Language identification: Examining the issues, *Proc. Symp. Document Analysis and Information Retrieval* (1996), pp. 125–135.
19. G. Botha, V. Zimu and E. Barnard, Text-based language identification for the South African languages, *Proc. 17th Ann. Symp. Pattern Recognition Association of South Africa*, Parys, South Africa (2006), pp. 7–13.
20. D. Benedetto, E. Caglioti and V. Loreto, Language trees and zipping, *Phys. Rev. Lett.* **88** (2002) 48702.
21. G. Windisch and L. Csink, Language identification using global statistics of natural languages, *SACI'05* (2005).
22. P. McNamee, Language identification: A solved problem suitable for undergraduate instruction, *J. Comput. Small Coll.* **20** (2005) 94–101.
23. C. Biemann and S. Teresniak, Disentangling from babylonian confusion-unsupervised language identification, *Proc. Computational Linguistics and Intelligent Text Processing (CICLing'05)*, Mexico City (2005), pp. 762–773.
24. A. Xafopoulos, C. Kotropoulos, G. Almpanidis and I. Pitas, Language identification in web documents using discrete hmms, *Pattern Recog.* **37** (2004) 583–594.
25. B. Hughes, T. Baldwin, S. Bird, J. Nicholson and A. MacKinlay, Reconsidering language identification for written language resources, *Proc. 5th Int. Conf. Language Resources and Evaluation (LREC'06)*, Genoa, Italy (2006), pp. 485–488.
26. J. Hakkinen and J. Tian, N-gram and decision tree based language identification for written words, *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU'01* (2001), pp. 335–338.
27. L.-F. Zhai, M.-H. Siu, X. Yang and H. Gish, Discriminatively trained language models using support vector machines for language identification, *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop* (2006), pp. 1–6.
28. N. Ljubesic, N. Mikelic and D. Boras, Language identification: How to distinguish similar languages? *Proc. 29th Int. Conf. Information Technology Interfaces*, Cavtat/Dubrovnik, Croatia (2007), pp. 541–546.
29. B. Martins and M. J. Silva, Language identification in web pages, *Proc. 2005 ACM Symp. Applied Computing* (2005), pp. 764–768.
30. P. Newman, Foreign language identification — A first step in translation, *Proc. 28th Ann. Conf. American Translators Association* (1987), pp. 509–516.
31. E. Giguet, The stakes of multilinguality: Multilingual text tokenization in natural language diagnosis, *Proc. 4th Pacific Rim Int. Conf. on Artificial Intelligence Workshop Future Issues for Multilingual Text Processing*, Cairns, Australia (1996).
32. C. Souter, G. Churcher, J. Hayes, J. Hughes and S. Johnson, Natural language identification using corpus-based models, *Hermes J. Linguistics* **13** (1994) 183–203.
33. N. C. Ingle, A language identification table, *The Incorporated Linguist* **15** (1976) 98–101.

34. R. D. Lins and P. Goncalves, Automatic language identification of written texts, *Proc. 2004 ACM Symp. Applied Computing*, ACM, Nicosia, Cyprus (2004), pp. 1128–1133.
35. W. B. Cavnar and J. M. Trenkle, N-gram-based text categorization, *Proc. 3rd Ann. Symp. Document Analysis and Information Retrieval*, Las Vegas, Nevada, USA (1994), pp. 161–175.
36. T. Dunning, Statistical identification of language, Technical Report CRL MCCS-94-273, Computing Research Lab (CRL), New Mexico State University (1994).
37. K. R. Beesley, Language identifier: A computer program for automatic natural-language identification of on-line text, *Proc. 29th Ann. Conf. American Translators Association* (1988), pp. 47–54.
38. J. Tian and J. Suontausta, Scalable neural network based language identification from written text, *Proc. 2003 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Vol. 1 (2003), pp. I–48–51.
39. M. J. Embrechts and F. Arciniegas, Neural networks for text-to-speech phone recognition, *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Vol. 5 (2000), pp. 3582–3587.
40. A. Selamat, C.-C. Ng and S. N. A. Ibrahim, Arabic script web document language identification using neural network, *Proc. 9th Int. Conf. Information Integration and Web Based Applications and Services* (2007), pp. 329–338.
41. A. Selamat and C.-C. Ng, Arabic script language identification using letter frequency neural networks, *Int. J. Web Information Systems* **4** (2008) 484–500.
42. Y. Yang, An evaluation of statistical approaches to text categorization, *Information Retrieval* **1** (1999) 69–90.
43. C.-C. Ng and A. Selamat, Improved letter weighting feature selection on Arabic script language identification, *Proc. 1st Asian Conf. Intelligent Information and Database Systems*, IEEE, Dong Hoi, Vietnam (2009), pp. 150–154.
44. D. Isa, V. Kallimani and H.-L. Lam, Using the self organizing map for clustering of text documents, *Expert Systems with Applications* **36** (2009) 9584–9591.
45. K. Saeed and M. Albakoor, Region growing based segmentation algorithm for typewritten and handwritten text recognition, *Applied Soft Computing* **9** (2009) 608–617.
46. M.-A. Fattah and F. Ren, GA, MR, FFNN, PNN and GMM based models for automatic text summarization, *Comput. Speech Lang.* **23** (2009) 126–144.
47. J. Wang, Y. Wu, X. Liu and X.-Y. Gao, Knowledge acquisition method from domain text based on theme logic model and artificial neural network, *Expert Systems with Applications* (In press, 2009).
48. F. Sebastiani, Machine learning in automated text categorization, *ACM Computing Surveys (CSUR)* **34** (2002) 1–47.
49. C. Cortes and V. Vapnik, Support vector networks, *Mach. Learn.* **20** (1995) 273–297.
50. T. Joachims, C. Nedellec and C. Rouveirol, Text categorization with support vector machines: Learning with many relevant, *Machine Learning: ECML'98 10th Euro. Conf. Machine Learning* (Springer, Germany, 1998).
51. S. Sagiroglu, U. Yavanoglu and E. N. Guven, Web based machine learning for language identification and translation, *Proc. 6th Int. Conf. Machine Learning and Applications* (2007), pp. 280–285.
52. S. MacNamara, P. Cunningham and J. Byrne, Neural networks for language identification: A comparative study, *Info. Process. Manag.* **34** (1998) 395–403.
53. M. Thompson, British broadcasting corporation (bbc) (2008), accessed June 2008.
54. G. Grefenstette, Comparing two language identification schemes, *Proc. 3rd Int. Conf. Statistical Analysis of Textual Data (JADT'95)* (1995).