Automatic Tamil Content Generation

S.Kohilavani, T.Mala and T.V.Geetha
Department of Computer Science & Engineering, Anna University
College of Engineering, Guindy, Chennai, India
kohila25@yahoo.com, mala@cs.annauniv.edu, rctamil@annauniv.edu

***Abstract* - Automatic content generation aims on developing an intelligent tutoring system in tamil language. This system focuses on delivering personalized content in Tamil language to an individual user needs based on their learning abilities and interests. This paper deals with automatic classification of Tamil documents and also the information extraction from those documents to construct the knowledge base. Documents are repositories of knowledge. There are numerous documents available and effective search in documents is time consuming. To make document search a simpler task, we need to perform document categorization. Document category can be found out using various techniques. In this paper, Naive Bayes (NB) algorithm is used to classify Tamil documents to one of pre-defined categories. Experiments are used to evaluate the Naive Bayes categorizer. The experimental results show that the Naive Bayes classifier performs well and its effectiveness is achieved with 89.8% accuracy. Informational words or sentences of the documents are then extracted using heuristic rules to fill up the predefined templates. An individual user's interests are identified and recorded to create a user profile. A user profile is specific to a user and is subjected to change over time. The Topic categorizer is used to categorize the topic based on user's query. The topic analyzer is used to analyze the user's profile and evaluate the user's knowledge using intelligent evaluator system. Based on the user's knowledge, intelligent evaluator system makes a decision to suggest the Location or to suggest a new topic retrieved from the knowledge base. Then the personalized content will be generated based on the knowledge level of the user. The experimental results show that the Content Generator performs well and its effectiveness is achieved with 82.35% accuracy.**

***Keywords* - Document Categorization, Naïve Bayes, Stopwords, preprocessing, classifier, information extraction.**

I. Introduction

Automatic Content Generation has always been an important application and research topic since the inception of digital documents. Today, document categorization is a necessity due to the very large amount of text documents that we have to deal with daily. A text categorization system can be used in indexing documents to assist information retrieval tasks. Automatic document categorization attempts to replace and save human effort required in performing manual categorization. It consists of assigning and labeling documents using a set of predefined categories based on document contents. Automatic text categorization has been used in search engines, digital library systems, and document management systems [13]. Barq for instance uses automatic categorization to provide similar documents feature [11]. In this paper, Naive Bayes which is a statistical machine learning algorithm is used to learn to classify Tamil text documents. The manual extraction of important knowledge from the tourism archive is a daunting task and does not guarantee that all main topics of interest will be discovered. In this paper, heuristic rules are used to extract the informational words or sentences to fill up the predefined templates. From the template, the personalized content was generated based on the knowledge level of the user. Knowledge level of the user was evaluated using intelligent evaluator system.

This paper is organized as follows. Section 2 briefly describe related works in the area of automatic text categorization and information extraction. Section 3 describes the system architecture. Section 4 describes the architecture of document categorization. Section 5 describes the preprocessing undergone by documents for the purpose of categorization; it describes in particular the preprocessing specific to the Tamil language. In section 6 Naïve Bayes (NB), the learning algorithm used in this paper for document categorization is presented. Section 7 describes the information extraction. Section 8 describes the Content Generation. Section 9 outlines the experimental setting, as well as the experiments carried out to evaluate the performance of the NB classifier and the content generation. Section 10 summarizes the work and suggests some ideas for future works.

II. Related works

The bulk of the text categorization work has been devoted to cope with automatic categorization of English and Latin character documents. El-Kourdi, M., A. Bensaid, and T. Rachidi used Naive Bayes algorithm to automatically classify non-vocalized Arabic web documents to one of five pre-defined categories [2]. Maria-Luiza Antonie and Osmar R. Zaiane proposed a novel approach for automatic text categorization that borrows from market basket analysis techniques using association rule mining in the data-mining field [9]. Wai Lam, Miguel Ruiz and Padmini Srinivasan developed an automatic text categorization approach and investigated its application to text retrieval. The categorization approach is derived from a

978-1-4244-4711-4/09/$25.00 ©2009 IEEE

combination of a learning paradigm known as instance-based learning and an advanced document retrieval technique known as retrieval feedback [12]. Thorsten Joachims introduced support vector machines for text categorization [7]. The present work evaluates the performance of the Naïve Bayes algorithm (NB) on Tamil documents. Chinglai Hor, Peter A. Crossley, Dean L. Millar suggests the use of a hybrid RS-GA method to process and extract implicit knowledge from operational data derived from relays and circuit breakers [1]. S.Iiritano and M.Ruffolo described a prototype of a vertical corporate portal that implements a Knowledge Discovery and Data Mining (KDD) process for knowledge extraction from unstructured data contained in textual documents [5]. Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal,Wendy Hall, Paul H. Lewis, and Nigel R. Shadbolt developed a tool called Artequakt. Artequakt automatically extracts knowledge about artists from the Web, populates a knowledge base, and uses it to generate personalized biographies [4].

III. System Architecture

The overall system architecture as shown in the Fig. 1 explains the various functionalities required to perform the automatic content generation. Automatic content generation focuses on delivering personalized content in Tamil language to an individual user needs based on their learning abilities and interests. Knowledge base is constructed using document categorization and tourism related information extraction.

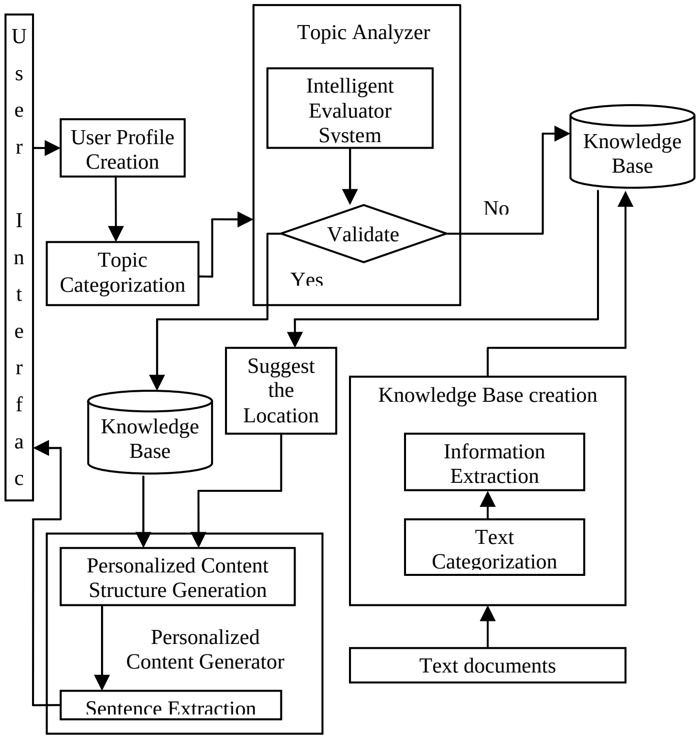

Figure 1. System Architecture

Text categorization stage involves preprocessing and classification process.The Tamil tourism documents are given as input to the preprocessing phase. In this phase the stop words are removed from each and every Tamil documents. These documents were used as the training set to build the classifier. After the stop words are removed, relevant words should be found out from the training set of documents and then passed to the classification phase. This phase uses naive bayes classification algorithm to classify the tourism documents as temples, hill stations and hotels.

After the classification process, information should be extracted from those documents. The extracted information should fill the template of each document of a particular category. Thus the knowledge base was constructed.

An individual user's interests are identified and recorded to create a user profile. A user profile is specific to a user and is subjected to change over time. The Topic categorizer is used to categorize the topic based on user's query. The topic analyzer is used to analyze the user's profile and evaluate the user's knowledge using intelligent evaluator system. Based on the user's knowledge, intelligent evaluator system makes a decision to suggest the Location of the spot or to suggest a new topic retrieved from the knowledge base. Then the personalized content will be generated based on the knowledge level of the user.

IV. Document Categorization

Document categorization is the task of automatically sorting a set of documents into categories from a predefined set. In the training phase, a set of Tamil tourism documents are given with class labels attached, and a classification system is built using a learning method. Once the categorization scheme is learned, it can be used for classifying future documents.

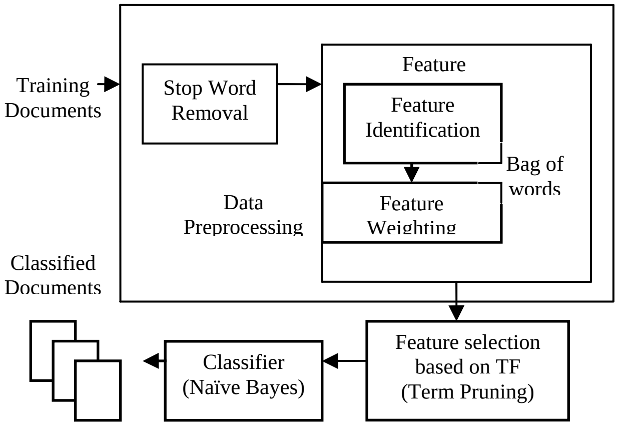

Figure 2. Detailed block diagram of Document categorization

Before a set of documents can be presented to a machine learning system, preprocessing should be done in each document. The document categorization as shown in Fig. 2 performs the categorization task by preprocessing and then classifies the documents based on naïve bayes algorithm.

The following sections gives a detailed view of preprocessing, and classification of documents.

V. Preprocessing

A data preprocessing phase is required to weed out those words that are of no interest in building the classifier and also to reduce the processing time. Preprocessing phase comprises of two stages namely stop word removal and term pruning.

A. *Stop word Removal*

The frequently occurring words are removed using stop word removal. The stop words are frequently occurring words whose content value is very low. In order to remove the stop words from the document, stop word list is prepared and the input tourism document is compared with the list and then stop words are removed.

B. *Term Pruning*

After removing the stop words, each document must be transformed in to a feature vector. Typically, each element of a feature vector represents a word from the document. The feature values should be integers indicating measure of frequency of the word's appearance in the text document. This text representation, referred to as the bag-of-words. The problem of finding a "good" subset of features is called feature selection. Feature selection is done by term pruning method. Term pruning is done according to the term frequency values. Here feature selection refers to relevant words. Relevant words are those which occurred more than 7 times but less than 26 times in the training set of documents. This would eliminate the very rare words that occurred and also the very common words that occur in almost every text document.

VI. Classification

Relevant words are then passed to the classification phase. This phase uses naive bayes classification algorithm to classify the tourism documents as temples, hill stations and hotels.

A. *The Classifier Module*

The classifier module is considered to be the core component of the document categorizer. It is responsible for classifying given Tamil documents to their target class. This is performed using the Naive Bayes (NB) algorithm. The NB classifier computes a posteriori probabilities of classes, using estimates obtained from a training set of labeled documents. When an unlabeled document is presented, the posteriori probability is computed for each class using (1); and the unlabeled document is then assigned to the class with the largest a posteriori probability.

1) *A Posteriori Probability Computation:*

- Let D be a document represented as a set of finite terms D={w1, w2, w3}.
- Let C be the number of target classes.
- Let docsi be the number of documents in category Ci and |Examples| be the number of documents in the training set of labeled documents.
- Let n be the total number of distinct stems in Ci
- Let Nk be the number of times wk occurs in Ci
- Then the a posteriori probability as given by Bayes theorem is:

$$P(Ci|D)=[P(Ci)*P(D|\ Ci)]/P(D).\ i=1,2,...C \qquad (1)$$

- When comparing a posteriori probabilities for the same document D, P(D) is the same for all categories and will not affect the comparison.
- The other quantities in (1) are estimated from the training set using NB learning.
- The assigned class AC(D) to document D is the class with largest a posteriori probability given by (1):

$$AC(D)=\text{argmax}Ci\ \{\ P(Ci|D).\ i=1,2,...C\}$$

B. *The Learning module*

The main task of the learning module is to learn from a set of labeled documents with predefined categories in order to allow the categorizer to classify the newly encountered documents *D* and to assign them to each of the predefined target categories Ci. The learning module is one way of estimating the needed quantities in (1) by learning from a training set of documents. Equation (3) gives an estimate for P(wk/Ci). Various assumptions are needed in order to simplify Equation (1), whose computations are otherwise expensive. These assumptions are applied to obtain the needed quantities for the class-conditional probabilities (Equations (4) and (5)).

1) *Naive Bayes learning algorithm:*

- Let D be a document represented as a set of finite terms/roots D={w1, w2,..., wn}.
- Let docsi be the number of documents in category Ci, and |Examples| be the number of documents in the training set of labeled documents.
- Step 1: collect the vocabulary, which is defined as the set of distinct words in the whole training set
- Step2: For each category Ci do the following
  - Compute P(Ci) = | docsi |/|Examples| (2)

where docsj is the number of training documents for the category is Cj.

- For each root wk in Vocabulary
  - Compute $P(wk/Ci)= (Nk,i +1)/(ni +| Texti |)$ (3)

  where Nk,i is the number of times wk occurs in Ci, ni is the total number of distinct terms in all training documents labeled Ci, and Texti is a single documents generated by concatenating all the training documents for category Ci .
- Equation (2) and (3) make use of the following two assumptions:
  - Assuming that the order of the words in a document does not affect the classification of the document:
    - $P(D|Ci)=P(\{w1,w2,...,wn\}|Ci)$ (4)
  - Assuming that the occurrence of each word is independent of the occurrence of other words in the document then:
    - $P(w1,...,wn|Ci)=P(w1|Ci)*P(w2|Ci)*... ………*P(wn|Ci)$ (5)

VII. INFORMATION EXTRACTION

Information extraction is a type of information retrieval whose goal is to automatically extract structured information from unstructured machine-readable documents. Fig.3 performs the information extraction task by template designing and using linguistic based heuristic rules to fill the templates.

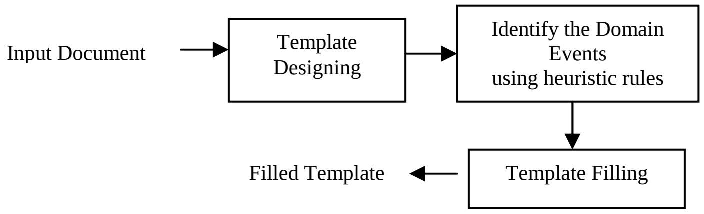

Figure 3. Detailed block diagram of Information Extraction

The focus in this work is the use of appropriate Tamil language processing techniques to extract relevant information from a set of categorized Tamil documents. Predefined templates that reveal the relevant information about the places have been designed to represent the documents. Linguistic based heuristic rules are used to extract the informational words from the Tamil documents and then the slots of the templates are filled with those informational words. Some of the Heuristic rules for the tourism domain are:

1. “இல் அமைந்துள்ளது”, “இல் அமைந்திருப்பது”, “இல் உள்ளது”, gives information about the location.
2. “பெருமை சேர்க்கிறது”- gives information about the special things about the spot.
3. “கொண்டாடப்படுகிறது”- gives infomation about the festivals
4. “இருந்து” - gives infomation for the slot “from”
5. “அழைக்கப்பட்டது”, “அழைக்கப்படுகிறது”, “அழைக்கப்படுகின்றது.”- denotes the othemames of the spot.
6. “போன்ற இடங்களையும்” – denotes the nearby places.
7. “ரயிலில் ”, “ரயில் மார்கமாக”- gives information about the places to reach the tourist spot by train
8. “சாலை மார்கமாக”, “சாலை”- gives information about the places to reach the tourist spot by bus
9. “கிலோமீட்டர்”- gives the number
10. “மேலும்”,“தொடர்ந்து” - gives additional infomation.

Corresponding slot should be filled with the help of the above rules. Fig.4 shows the filled template for temple category.

Location of the spot	:	திருச்செந்தூர்
God name	:	முருகன்
Religion	:	இந்து
From	:	திருநெல்வேலி
Km	:	60
By Train	:	சென்னை-திருநெல்வேலி
By Bus	:	மதுரை, திருச்சி, சென்னை, கோவை

Figure 4. Filled template for temple category.

VIII. CONTENT GENERATION

An individual user’s interests are identified and recorded to create a user profile. User profile should consist of user id, password, category of the document viewed, Level of the document viewed, time and date. A user profile is specific to a user and is subjected to change over time. The topic analyzer is used to analyze the user’s profile and evaluate the user’s knowledge using intelligent evaluator system. Topic Analyzer includes Evaluator and validator. Validator raise the questions about the tourist spot and determines the knowledge level. For example, if the user is in need of a particular spot information, the Evaluator makes the Validator to raise the questions based on the level of clearance in the user profile. Knowledge level can be classified in to 3 types.

Level 1 → not known about the Location of the spot.
Level 2 → known about the spot to some extent.

Level 3 → known about the spot.

If the level of clearance is 0, the Validator raise the questions of level 1. If the user cleared level 1, then the level and status of the user profile will be updated and the Validator goes to next level. Otherwise, the information about the location of the tourist spot will be provided to the user. If the user cleared level 2, that is, the user known about the spot to some extent, then the Validator goes to next level and update the user profile. Otherwise the personalized content of some details about that tourist spot will be provided to the user. If the user cleared level 3, that is, the user known about the spot, then the Validator displays the congrats message and also update the user profile. Otherwise the personalized content of the detailed information about that tourist spot will be displayed. The following Fig.5 shows the detailed design of Personalized Content Generator.

Once the topic was analyzed from the topic analyzer, corresponding level of the template should be extracted from the knowledge base based on the knowledge level of the user and the corresponding document should be extracted from the corpus.

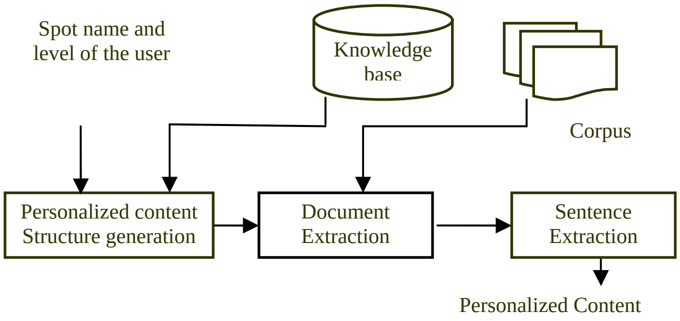

Figure 5. Detailed design of Personalized Content Generator.

Finally, sentence should be extracted from the same document by matching the slots in the template. Thus the content was generated from the corpus based on the knowledge level of the user. The following section shows the performance evaluation of the document categorization and personalized content generation.

IX. Performance Evaluation And Results

A. *Corpus Collection*

The evaluation experiment is done on Tamil text documents. These documents were the collection of Tamilnadu tourism documents. From the 50 documents, the first 30 are used for training and the second 20 are used for testing. The classification task considered here is to assign the documents to one category of the two categories.

B. *Experiment Results for Document Categorization*

The Precision/Recall is used as a measure of performance for document categorization. Recall is the percentage of total documents for the given topic that are correctly classified. Precision is the percentage of predicted documents for the given topic that are correctly classified. Given a test set of N documents, a two-by-two contingency table with four cells can be constructed for each binary classification problem. The cells contain the counts for true positive (TP), false positive (FP), true negative (TN) and false negative (FN), respectively. Here, TP (True Positives) is the number of documents assigned correctly to class i. FP (False Positives) is the number of documents that do not belong to class i but are assigned to class i incorrectly by the classifier; and FN (False Negatives) is the number of documents that are not assigned to class i by the classifier but which actually belong to class i. The terms used to express precision and recall are given in the contingency table as shown in Table 1. For evaluating the effectiveness of the system, F-measure is used. A popular measure that combines Precision and Recall is the weighted harmonic mean of precision and recall. The metrics for binary-decisions are defined as:

- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 = 2 * Recall * Precision / (Recall + Precision)

TABLE I

Contingency Table For Category Kovil

Category **Kovil**		**Human assignments**	
		Yes	***No***
Classifier **Assignments**	***Yes***	TP	FP
	No	FN	TN

To evaluate overall performance across the entire set of categories, the results are macroaveraged, i.e. by taking the average of F-measure values for each category and the overall average is computed. The recall and the precision measures are calculated for each category and the average values are shown in the following Table 2.

TABLE II

Performance Evaluation Results Table

Category	**Average Precision**	**Average Recall**	**Average F-measure**
Kovil	100.00	80.00	88.89
Malai	83.33	100.00	90.71
Macroaverage	91.67	90.00	**89.80**

The result shows that the Naïve Bayes achieves a macroaverage of 89.8. Thus it shows that the naïve bayes classifier performs well and its effectiveness is better in classifying the Tamil documents. The Fig. 6 shows a pictorial representation of the number of documents correctly classified and the number of documents misclassified in each category.

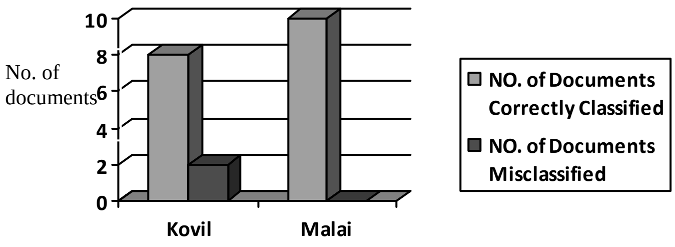

Figure 6. Graphical Representation

The above graph shows that out of 20 testing documents, 8 documents are correctly classified as Kovil category and 2 documents are misclassified and the remaining 10 documents are correctly classified as Malai category.

C. *Experiment Results for Personalized Content Generation*

The Precision/Recall is used as a measure of performance for personalized content generation. Recall is the percentage of total documents for which the personalized content are generated correctly. Precision is the percentage of predicted documents for which the personalized content are generated correctly.

TABLE III

PERFORMANCE EVALUATION RESULTS TABLE

Technique	**Average Precision**	**Average Recall**	**Average F-measure**
Personalized Content	100.00	70.00	82.35

The recall and the precision measures are calculated and the average values are shown in the Table 3.

The result shows that the Automatic content generation achieves an average F-measure of 82.35. Thus it shows that this system performs well and its effectiveness is better in generating the content for Tamil documents.

X. CONCLUSION AND FUTURE WORK

This work has been carried out to automatically classify Tamil documents using the NB algorithm, with the use of a different data set and a different number of categories and also the information was extracted from those documents to construct the knowledge base. After the categorization was done, tourism related information was extracted from those documents. An individual user's interests are identified and recorded to create a user profile. The topic analyzer analyzed the user's profile and evaluated the user's knowledge using intelligent evaluator system. Once the topic was analyzed, Personalised content was generated from the corpus. The experimental results show that the Naïve Bayes classifier performs well and its effectiveness is achieved with 89.8% accuracy and also the experimental results show that the Personalized content generation performs well and its effectiveness is achieved with 82.35% accuracy. Advanced TF-IDF techniques can be deployed in the next level for categorizing the documents. The current work is confined to tourism documents of Tamilnadu state, but it can be extended to all states in India. And also the current work is confined to Tamil text documents, it can also be further extended to all Indian languages. Categorization work can be further enhanced by adding more categories.

REFERENCES

[1] Chinglai Hor, Peter A. Crossley, and Dean L. Millar, "Application of Genetic Algorithm and Rough Set Theory for Knowledge Extraction," IEEE transactions on Power Tech, pp. 1117 - 1122 , July 2007.

[2] M. El-Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages, pp. 51-58, August 2004.

[3] Fabrizio sebastiani, "Machine Learning in Automated TextCategorization," ACM Computing Surveys, Vol. 34, Issue No. 1, pp. 1–47, March 2002.

[4] Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel R. Shadbolt, "Automatic Ontology-Based Knowledge Extraction from Web Documents," IEEE transactions on Intelligent systems, Vol.18, Issue no.1, pp-14-21, January 2003.

[5] S.Iiritano, and M.Ruffolo,"Managing the Knowledge Contained in Electronic Documents: a Clustering Method for Text Mining," Proceedings of 12th International Workshop on Database and Expert Systems Applications, pp. 454 – 458, September 2001.

[6] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," Proceedings of 14th International Conference on Machine Learning, 1997.

[7] Joachims Thorsten, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proceedings of 10th European Conference on Machine Learning, pp. 137-42. 1998.

[8] Lewis, and M. Ringnette, "Comparison of two learning algorithms for text categorization," Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval, 1994.

[9] Maria-Luiza Antonie and Osmar R. Zaiane, "Text document categorization by term association," Proceedings of IEEE International Conference on Data Mining, pp.19 – 26, December 2002.

[10] M.Sahami, "Learning limited dependence Bayesian classifier,"Proceedings of the second international Conference on Knowledge Discovery and Data Mining, pp.335-338, AAAI press, 1996.

[11] T. Rachidi, O. Iraqi, M. Bouzoubaa, A. Ben AlKhattab, M. El Kourdi, A. Zahi, and A. Bensaid, "Barq: distributed multilingual Internet search engine with focus on Arabic language," Proceedings of IEEE Conference on Sys., Man and Cyber., Washington DC, October 5-8, pp. , 2003.

[12] Wai Lam, M. Ruiz, and P. Srinivasan, "Automatic text categorization and its application to text retrieval," IEEE Transactions on Knowledge and Data Engineering, Vol. 11, Issue no. 6, pp. 865 – 879, November 1999.

[13] Y. Yang, "An evaluation of statistical approaches to text categorization," Journal of Information Retrieval, Vol. 1, Number 1-2, pp. 69--90, 1999.