Improved Reinforcement-based Profile Learning for Document Filtering

Md. Nasir Sulaiman, Yahya M. Al Murtadha, Zaiton Muda and Aida Mustapha

University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia

Summary
A personalized information filtering system tailors user queries to the current user interests and adapt the information as they change over time. The system monitors a stream of incoming documents to learn user information needs in the form of profiles and yield relevant documents filtered to only those matches the user profiles. To learn the profile, the significance of query terms will be accessed and weights will be assigned to each term in the profile. This article proposed purity terms weighting method for profile learning in a personalized information filtering system. The main idea is to weigh the terms based on their pure frequencies, in addition to the number of pure relevant documents that contain them. The profiles are discriminated based on top weighed terms that represent the profiles. Profiles are also updated with every selected relevant document in order to match user interests. The efficiency of the proposed method is measured by using linear utility accuracy tested on TREC 2002 filtering track. The experimental results show improvement in terms selection and profile building accuracy as compared with Rocchio's Algorithm, Okapi/BSS Basic Search System, and the incremental profile learning approach.

Key words:
Information Filtering, User Profile Learning, Term Weighting, Reinforcement Learning, TREC 2002.

1. Introduction

There are numerous text documents available in electronic form, while more and more are becoming available every day. Such documents represent a massive amount of information. The escalating number of information containers, users, and their increasing demands for the information has caused the retrieval of huge and irrelevant information. In addition, the information sources set a dynamic and unorganized environment where the information appear and disappear at any time. Gathering information from such environment is similar to drinking water from a fire hose, metaphorically. Hence, there are many occasions when novice users are not able to get the information they require. Seeking values in this huge collection requires organization and much of the work of organizing documents can be automated through text classification [1]. Information filtering through intelligent and personalized system remains a challenge [2].

To alleviate this problem, Information Filtering (IF) and user profiling is introduced. IF is a field of study designed for creating a systematic approach 0to extracting information that a particular person finds important from a larger stream of information. A text filtering system sifts through a stream of incoming information to find documents relevant to a set of user needs represented by profiles. IF monitors a stream of incoming document to find those match the user's information need called profile. Profiles are the basis of the performance of IF systems. Filters are tools to help people find the most valuable information, so that the limited time spent on locating the information can be maximized on finding the most interesting and valuable documents. Filtering differs from search in that documents arrive sequentially over time [3].

Nonetheless, the uncertainties that exist in filtering environments make it extremely difficult to gather and maintain accurate information necessary for filtering [4], due to the dynamic nature of the user's interests and the documents stream. To manage such uncertainties requires a high level of adaptivity on the system's part. This adaptivity can be achieved by applying various machine learning techniques. The overall problem of the IF may then be interpreted as learning a map from a space of documents to the space of real–valued user relevance factors. The capability of model to learn user's preferences is at the heart of a personalized information filtering system. The main challenge with existing personalized filtering lies in building the user profiles, which is how to measure and select the most suitable terms (attributes) that can help to discriminate between the filtered classes and the learned user interests.

The construction of accurate profiles is a key task, whereby the system's success will depend to a large extent on the ability of the learned profile to represent the user's actual interest [5]. All information related to the user has to

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

be collected in a unified user profile. Profile learning is essential in the filtering process, it aims to collect the user preferences on a unified profile and match the incoming documents against this profile. The filtering system should block the delivery of the documents that the user is less likely to be interested in. To identify whether a document is relevant or not, a score that measures the similarity between the document and the profile is computed. When the score is higher than the similarity threshold then the document is selected, otherwise the document is rejected.

2. Related Works

Profile Learning has been studied by two communities, Information Filtering and Machine Learning [6]. Most of the researches in IF use an incremental version of Rocchio's algorithm [7, 8]. Subsequent researchers proposed a number of profile learning methods based on Rocchio's algorithm. Microsoft Research Laboratory in Cambridge has developed an evaluation environment called Keenbow for a wide range of Information Retrieval (IR) experiments. One component in Keenbow is Okapi/BSS [9] which uses terms weighting in addition to the Query expansion. The incremental profile learning based on reinforcement method [6] is an IF profile learning method based on the terms weighting. The method weighs the terms according to the frequency of term occurrence in the documents and the profiles, without considering the correlation with the pure occurrence in either the relevant or irrelevant documents.

2.1 Rocchio's Algorithm

A feedback query creation algorithm developed by Joe Rocchio in the mid-1960's has, over the years, proven to be one of the most successful profile learning algorithms. Rocchio's algorithm was developed in the framework of the Vector Space Model (VSM). Sebastian [10] developed a range of classifier that ranged from fast (Rocchio's) to the accurate with huge data. The algorithm is based on the assumption that if the relevance for a query is known, an optimal query vector would maximize the average query-document similarity for the relevant articles; while simultaneously minimize the average query-document similarity for the non-relevant documents. Rocchio has shown that an optimal query vector is the difference vector of the centroid vectors for the relevant and the non-relevant articles:

$$\vec{Q}_{new} = \alpha \times \vec{Q}_{orig} + \beta \times \frac{1}{R} \sum_{D \in \mathrm{Re}l} \vec{D} - \gamma \times \frac{1}{N-R} \sum_{D \notin \mathrm{Re}l} \vec{D}$$

where R is the number of relevant articles, $\vec{D}$ is the documents terms vector and N is the total number of articles in the collection, α, β, γ are coefficients introduced by Rocchio to control the original query. The feedback query created by Rocchio's query reformulation process using the documents marked relevant by a user is now considered as the user profile. New incoming documents are matched against this profile and are routed to the user if they have a suitable match to the profile. We compared our improved weighting method with the incremental version of Rocchio's algorithm described in query zoning by Singhal and his colleagues [8].

2.2 Query Expansion on Okapi/BSS Basic Search System

BSS is a weighting component of the evaluation environment Keenbow of IR developed by Microsoft Research laboratory. Robertson and Walker [11] introduced the query expansion in Okapi as follows. Given some relevant documents:

1. Extract all terms from all documents judged or assumed to be relevant,
2. Rank all terms in order of assign weights,
3. Select those terms that are above a threshold or cut-off value, defined as a threshold on the offer weight and/or cut-off on the number of terms, and
4. Weigh the terms according to the usual relevance weighting formula.

Since initial step is to choose the term, terms are ranked in decreasing order of the term selection value or offer weight and the top ranked terms are then chosen:

$$TSV = r \cdot w$$

The variable *r* is the number of known relevant documents in which the term occurs and *w* is the weight by Robertson/Sparck-Jones [12]:

$$w_i^t = \log\left(\frac{\left(r_i^t + 0.5\right) / \left(R^t - r_i^t + 0.5\right)}{\left(n_i^t - r_i^t + 0.5\right) / \left(N^t - n_i^t - R^t + r_i^t + 0.5\right)} \right)$$

where at time *t*, R^t is the number of documents known to be relevant to a specific topic, r_i^t is the number of relevant documents containing the term, n_i^t is the number of documents containing the term, N^t is the number of items (documents) in the collection. An absolute threshold criterion for the offer weight to select the best term is:

$$NTSV \succ c$$

This method does not considering the correlation with the term's pure occurrence in either the relevant or irrelevant documents.

2.3 Incremental Profile Learning Approach

Tebri and his colleagues [6] proposed an incremental learning based on reinforcement algorithm. It is processed each time a filtered document is judged as relevant by the user. The learning consists of adapting the profile by updating the term profile weights, adding new terms, and removing some non-significant terms. The adaptation rule works as follows. First of all, when a given document d is judged as relevant by the user, determine the best temporary profile p_x^t to select this document with a high score. This can be written as the following:

$$rsv\left(d^t, p_x^t\right) = \lambda$$

where λ is the desired relevance value RSV, d^t is the documents at time t and P_x^t is the profile at time t. This equation has many solutions, to overcome this problem; a constraint was proposed to be added in order to obtain a unique solution. To do so, they defined what they called the ideal profile and the ideal weight as follows:

Definition 1: The ideal profile at t instant is the profile that enables to select only relevant documents.

Definition 2: The ideal weight of a term is its weight in the ideal profile.

Taking into account the incremental aspect of the learning process, they considered the term weights, pw_i^t is the temporary weight of the temporary profile, p_x^t is calculated along with the ideal weight f_i^t and the documents weights d_k^t, while then integrated into the final profile. Nevertheless, the idea of the temporary profiling is time consuming since many computations are required.

3. The Proposed Term Weighting Approach

This article focuses on improved profile learning for learning user preferences and filtering documents. As mentioned previously, the profiles and documents are represented as vector of terms coupled with their associated weights based on the Vector Space Model (VSM). For each document that is entered into the system, it will first be indexed, represented, and tested for relevance similarity measurement. To judge whether a document is relevant to the user or otherwise, the system should extract the best terms from the collection of document terms that can help to classify the profile. The profile is then updated by adding new terms and removing non-significant terms at each selected document. To best choose the terms their weights or "degree of importance" should be computed.

3.1 Query Expansion (Term's Degree of Importance)

Under the VSM model, if a relevant document does not contain the terms that are in the query, then that document will not be retrieved. The aim of query expansion is to reduce this query or document mismatch by expanding the query to include words or phrases with similar meaning, or with some other statistical relations to the set of relevant documents. Lin and his colleagues in [13] presented a new method for query expansion based on user relevance feedback techniques for mining additional query terms. According to the user's relevance feedback, the proposed query expansion method calculates the degrees of importance of relevant terms of documents in the document database using the following formula:

$$\text{Importance} = \left(\frac{F_{purity} - \min_{i=1}^{n} F_{purity} + 1}{0.5 + \log_{10}\left(\frac{M}{F_{ri}}\right)} \right) \times \log_{10}\left(\frac{\left(F^*_{purity_i} - \min_{i=1}^{n} F^*_{purity_i} + 1\right)^2}{F^*_{ri}} \right) \quad (1)$$

where $F_{purity_i} = F_{ri} - F_{iri}$ and $F^*_{purity_i} = F^*_{ri} - F^*_{iri}$.

F_{ri} is the frequency of relevant term t_i appearing in relevant documents and F_{iri} is the frequency of relevant term t_i appearing in irrelevant documents. Meanwhile, F_{ri}^* denotes the frequency of relevant term t_i and the previous query terms appearing in relevant documents simultaneously; and F_{iri}^* is the frequency of relevant term t_i and the previous query terms appearing in the irrelevant documents simultaneously. M denotes the number of documents including relevant term t_i in the document database, while the value 0.5 is used to avoid the denominator to be zero.

One drawback of the query expansion is overfitting due to the presence of noise. Some terms including the

background noise terms can only be discriminated between the relevant and non-relevant feedback documents, but cannot be generalized to rank the relevancy of the remaining unlabeled documents.

3.2 Term's Purity Weighting Method

By analyzing the weighting formula in (1) to compute the importance of terms to a profile, we found out that this method only considers the frequency of the terms in the relevant and irrelevant documents. The question now is: "In how many relevant documents this term occurs?"

Consider the following situation. For a term t_i; there are 50 relevant documents and 50 irrelevant documents. Assuming the frequency of term t_i in the relevant documents is 70 times and the frequency of this term t_i in the irrelevant documents is 50 times. Using the formula in (1), we can compute the importance of the term t_i in relation to their frequencies. However, these frequency values may be in one, two or just few documents among the documents collections, which mean some documents may be more dominant in term of terms weighting, selecting, and profile building as compared to other documents.

To investigate the correlation between term frequency and pure number of relevant documents, we propose a term purity weighting method to compute the term's purity occurrences in each profile. This method is computes the term importance through the correlation between the term frequency in the documents and the number of relevant documents that the term occurs. The weighting formula is as follows:

$$TermPurity = \left(\frac{D\ purity_i - \min_{i=1}^{n} D\ purity_i + 1}{0.5 + Log_{10}\left(\frac{M}{F_{ri} + 0.1}\right)} \right) \times \log_{10} \left(\frac{\left(F\ purity_i - \min_{i=1}^{n} F\ purity_i + 1 \right)^2}{Fr_i + 0.1} \right) \quad (2)$$

where $D_{purity_i} = dn - dr$ and $F_{purity_i} = F_{ri} - F_{ri_i}$.

The variable dn is the number of all relevant documents including the term t_i, dr is the number of selected relevant document including the term t_i, F_{ri} denotes the frequency of relevant term t_i appearing in relevant documents, and F_{iri} is the frequency of relevant term t_i appearing in irrelevant documents. The value 0.1 is used to avoid the denominator to be zero. This value should be small to as 0.1 to modify the calculation of the term purity to get as small weighting as possible. This is due to the fact that the relevance calculation should be within the range of zero and one.

4. Methodology

The aim of the experiments is to evaluate effectiveness of the proposed scheme on profile learning based on *Reuters* dataset provided by TREC. In this section we describe our methodology and learning algorithms.

4.1 Dataset Collection

The *Reuter* dataset is provided by National Institute of Standard and Technology (NIST) for research purposes. We use the Filtering Track data of TREC-2002 [3]. It consists of news stories covering the period of a year in 1996-1997. Items in the collection have unique identifiers and are dated but are not timed. The first 6-week items, 20 August through 30 September 1996, were taken as the training set. The remainder of the collection formed the test set. Topics R101-R110 from TREC 2002 are used as profiles and the relevance judgment of each topic is used to simulate the user judgment. The system processed one document a time. Figure 1 shows part of TREC 2002 filtering track document for topic R101.

<newsitem itemid="63261" id="root" date="1996-09-19" xml:lang="en">
<Title>USA: Economic espionage bill clears U.S. Senate.</title>
<headline>Economic espionage bill clears U.S. Senate.</headline>
<dateline>WASHINGTON 1996-09-19</dateline>
<text>
A bill to make the theft of U.S. trade secrets by foreign governments and companies a federal crime was heading for final congressional action after approval by the Senate.
< p>The Senate version of the bill, passed Wednesday night, must be reconciled with a different House version passed by the House Tuesday. Final action is expected by next week, a House Judiciary Committee staff member said.
Sponsors of the bill said U.S. companies were losing $24 billion a year from the theft of computer software, advanced technology and other intellectual property. They said current federal laws did not cover such crimes. "

Fig 1 A part of one TREC 2002 filtering track document for topic R101

4.2 The Learning Procedure

Data pre-processing is standard: terms were stemmed using the Porter Stemming and stop words were removed

by using standard stop word list. At the beginning of the process, the system should build the initial profile by extracting the terms from the title and the description of the profile. Since the title and the description are not enough for building the initial profile, we apply the concept of the pre-test documents applied by TREC 2002, where there are three documents for each profile that are assumed to be relevant. However, even a perfect adaptive profile updating mechanism could suffer a poor result if the update starts from a biased initial profile. In fact, there is a high potential to get a bias initial profile because of insufficient topic features provided by such few initial positive training documents [14]. For each profile, the overall process can be summarized as follows:

I. At $t = 0$, build the initial profile by extracting the terms from the title and the description of each profile. Use the three pre-test relevant documents, and then weigh the profile terms using equation [6] for each document.

$$w_i^0 = \frac{tfp_j}{\max_j\left(tfp_j\right)}$$

II. Each incoming document at time t is indexed, to build a list of stemmed terms [15]; the terms belonging to a stop list are removed. Then compute the weight for every term in the document using equation [12]:

$$d_i^t = \frac{t f_i^t}{h3 + h4 * \frac{d l^t}{\Delta l^t} + t f_i^t} * \log\left(\frac{N^t}{n^t} + 1\right)$$

where, tf_i^t is the frequency of the term t_i in the document d_t; $h3$ and $h4$ are constant parameters. For the purpose of experiments $h3$=0.2 and $h4$=0.7; d_l^t is the document length, and d_t is the number of index terms.

Δl^t is the average document length; N^t is the number of incoming documents until term t_i; and n^t is the number of incoming documents containing the term t_i.

III. Compute the relevance between the document and the profile using the equation [6]:

$$rsv\left(d^t, p^t\right) = \sum_{t_i \in d^t, tp_j \in pt \text{ and } t_i = tp_j} d_i^t \times w_j^t$$

where d_i^t is the term's weight in the document and w_i^t is the term's weight in the profile. If the value of rsv is greater than the threshold value, then the document is accepted, otherwise the document is rejected. A zero should be an adequate value for a threshold.
If the document RSV > threshold go to (IV), else go to (II).

IV. If the document is a user-relevant, do
   A. Weigh the terms using Purity Weighting Method (our proposed method).
   B. Select the highest 60 terms to for each profile sorted by their weights.
   C. Update the profile by combining the user profile with the selected terms , then for any existing term update its weight with the new one, finally select the highest 60 terms to be the new profile and remove the remaining terms from the profile. This profile will be used foe the filtering.

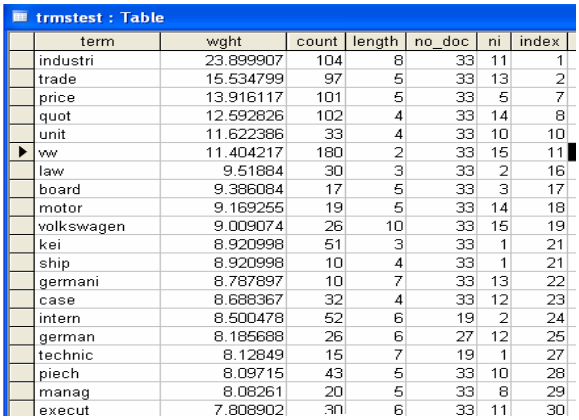
   Else, go to (II).

V. After learning all ten profiles, the output tables are used to filter the new documents. Each new document is indexed to be represented as a vector of words, and then Dice Measure [16] is computed between the document and the profile. Finally, the new document will be matched against the learned profile, and routed to the user once they have a suitable match to the profile. The value [0.3, 1] means the new document is relevant to the profile, otherwise the document is irrelevant.

5. Experimental Results

Based on the proposed method, a profile learning system is developed using Microsoft VC++ connected to Microsoft Access database through an Open Database Connection (ODBC). The output is a list of maximum 60 highest weighted terms for each profile in Microsoft Access table format. These tables represent the learned profiles, which are used to filter the incoming new documents.

Figure 2 shows the system output for profile R101 as a vector of terms, whereby each term is associated with its weight, length, frequency, and number of documents that contained the terms.



trmstest : Table

	term	wght	count	length	no_doc	ni	index
	industri	23.899907	104	8	33	11	1
	trade	15.534799	97	5	33	13	2
	price	13.916117	101	5	33	5	7
	quot	12.592826	102	4	33	14	8
	unit	11.622386	33	4	33	10	10
▶	vw	11.404217	180	2	33	15	11
	law	9.51884	30	3	33	2	16
	board	9.386084	17	5	33	3	17
	motor	9.169255	19	5	33	14	18
	volkswagen	9.009074	26	10	33	15	19
	kei	8.920998	51	3	33	1	21
	ship	8.920998	10	4	33	1	21
	germani	8.787897	10	7	33	13	22
	case	8.688367	32	4	33	12	23
	intern	8.500478	52	6	19	2	24
	german	8.185688	26	6	27	12	25
	technic	8.12849	15	7	19	1	27
	piech	8.09715	43	5	33	10	28
	manag	8.08261	20	5	33	8	29
	execut	7.808902	30	6	33	11	30

Fig 2 An example of R101 profile output database as vector of terms

The following Table 1 shows the weights of ten most significant terms of the profile R101 learned and sorted by our method. The weights are compared against actual weights learned by the Reinforcement Incremental Profile Learning method in [6] that utilized the same dataset. From Table 1, we show that our method assigns much higher values than the other methods for the most significant terms. The blank cells mean that the term is not selected by the Reinforcement method within the selected 60 terms.

Table 1 List of Weighted Terms for Profile R101 Sorted by Purity Method

Term	Purity	Reinf.
industr	23.899907	----------
motor	18.631943	12.6261
trade	15.534799	----------
price	13.916117	----------
quot	12.592826	----------
unit	11.622386	----------
vw	11.404217	9.110769
volkswagen	11.380277	19:7582
law	9.51884	2.935971
board	9.386084	----------

Table 2 shows the top-ten terms selected for the profile R101 sorted by Rocchio's method in comparison with Reinforcement and the proposed Purity method. Again, terms weights for Rocchio and Reinforcement methods are as published by Tebri and his colleagues in [6] in experiments using the same dataset. From the table, we can see that the Reinforcement method assigns higher values for some terms than our method do, which means that these terms seems to be less important to our method as compared to the Reinforcement method. It is clear from the tow tables that the different methods select different terms since the term's selection plays an important role in getting a better filtering result. As illustrated in Table 1, our method select many terms for the profile R101 that other methods do not select. These terms considered as the best terms to build the profile and match the incoming documents against them. This can be clear by calculating the filtering accuracy for each method.

Table 2: List of Weighted Terms for Profile R101 Sorted by Rocchio

Term	Rocchio	Reinf.	Purity
lopez	4.4116	22.6087	5.30018
volkswagen	3.7142	19.7582	11.380277
opel	3.4781	15.2014	4.797744
germany	3.3294	10.3350	12.436979
secret	3.2878	12.3694	7.710709
jose	3.1212	12.1792	----------
ignacio	3.0875	16.4970	----------
frankfurt	3.0829	08.8880	7.63702
motor	3.0795	12.6261	18.631943
automak	3.0211	10.5031	--------

The filtering accuracy is evaluated based on T11SU utility [3]. The evaluation is described by TREC as below. The particular parameters being used are a credit of 2 for a relevant document retrieved and a debit of 1 for a non-relevant document retrieved.

T11U = 2 * No. of relevant docs retrieved –
No. of non_ relevant docs retrieved

This corresponds to the retrieval rule,
retrieve if P(rel) > 0.33.

The utility for each topic is measured by:

$T11SU$ = max ($T11NU$, $MinNU$) - $MinNU$/ 1 – $MinNU$
$T11NU = T11U / MaxU$
$T11U$ = 2 × (No. of relevant docs retrieved) –
(No. of non-relevant docs retrieved)
$MaxU$ = 2 × (No. of relevant docs)
$MinNU$ = -0.5

Table 3 compares the average T11SU Utility obtained by the four methods on TREC topics which measure the average of the correctly retrieved documents against the entire dataset documents. It illustrates that our method precision average is 0.525 which is higher than the precision average of other compared learning methods.

Table 3: T11SU Evaluation for Four Methods on TREC Dataset

	Purity	Reinf.	Okapi/BSS	Rocchio
T11SU	0.525	0.462	0.354	0.427

6. Conclusions

The main purpose of the experiment is to learn user interests and to build user profiles that are able to yield high number of relevant documents based on the user profiles. The idea is to improve the existing term weighting scheme to select best terms representing user preferences (profiles) in order to help discriminating between profiles through analysis of document content rather than user behaviors. The results ahs proven that the proposed purity term weighting method yield higher accuracy performance in learning the user interest and building the profiles. This method can be improved and applied in misuse detection or the recommendation systems. In the near future, we are looking at investigating the dependency between the selected terms in individual user profiles.

References

[1] Rennie, J.D.M. 2001. Improving Multi-class Text Classification with Naive Bayes. Master Thesis, Massachusetts Institute of Technology (MIT).

[2] [2] Albayrak, S., Wollny, S., Varone, N., Lommatzsch, A., and Milosevic, D. 2005. Agent Technology for Personalized Information Filtering: The PIA System. In *Proceedings of the 2005 ACM Symposium on Applied Computing,* Santa Fe, New Mexico. ACM Press New York, USA, pp. 54–59.

[3] Robertson, S. E. and Soboroff, I. 2002. The TREC 2002 Filtering Track Report. In *Proceedings of the 11th Text REtrivel Conference (TREC-11)*, Santa Fe, New Mexico, USA. Department of Commerce, NIST Special Publication.

[4] Mostafa, J., Mukhopadhaya, S., Lam, W., and Palakal, M. 1997. A Multilevel Approach to Intelligent Information filtering: Model, System, and Evaluation. *ACM Transaction on Information Systems*. 15(4):368–399.

[5] Balabanovic, M. and Shoham, Y. 1997. FAB Content-based Collaborative Recommendation. *Communications of the ACM*. 40(3):66–67.

[6] Tebri, H., Boughanem, M. and Chrisment, C. 2005. Incremental Profile Learning based on a Reinforcement Method. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, Santa Fe, New Mexico, USA. Department of Commerce, NIST Special Publication, ACM Press.

[7] Salton, G. and Buckley, C. 1990. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*. 41(4):288–297.

[8] Singhal, A., Mitra, M. and Buckleytt,C. 1997. Learning Routing Queries in a Query Zone. In *Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia: ACM, pp. 25–32.

[9] Robertson, S. E. and Walker, S. 2000[a]. Okapi/Keenbow at TREC-8. In *Proceedings of the 8th Text REtrival Conference (TREC-8)*, Gaithersburg, Maryland, USA: NIST Special Publication, pp. 151–161.

[10] Shih, L. K. 2004. Machine Learning on Web Documents, PhD Thesis, Massachusetts Institute of Technology (MIT).

[11] Robertson, S. E. and Walker, S. 2000. Microsoft Cambridge at TREC-9, *Proceedings of the 9th Text REtrival Conference (TREC-9)* Gaithersburg, Maryland: NIST Special Publication.

[12] Robertson, S. E. and Sparck-Jones, K. 1976. Relevance Weighting of Search Terms. *JASIS*, 27(3):129–146.

[13] Lin, H.-C., Wang, L.-H., and Chen, S.-M. 2006. Query Expansion for Document Retrieval based on Fuzzy Rules and User Relevance Feedback Techniques. S*cience Direct*, 31(2):397–405.

[14] Ma, L., Chen, Q., Ma, S., Zhang, M., and Cai, L. 2002. Incremental Learning for Profile Training in Adaptive Document Filtering, *Eleventh Text REtrieval Conference (TREC 2002)*. Maryland, USA.

[15] Porter, M., 1979. An Algorithm for Suffix Stripping. *Program,* 14(3):130–137.

[16] Salton, G., Wong, B. A., and Yang, C. S. 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Boston, MA, USA Addison-Wesley Longman Publishing

Md. Nasir Sulaiman is an Associate Professor in the Dept. of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. He obtained PhD in Neural Network Simulation from Loughborough University, U.K. in 1994. His research interests include intelligent computing, software agents, and data mining.