

Personalized Optimal Search in Local Query Expansion

Shan-Mu Lin¹ and Chuen-Min Huang²

¹Dept. of Information Management, National Yunlin University of Technology and Science, Douliou, Taiwan

²Dept. of Information Management, National Yunlin University of Technology and Science, Douliou, Taiwan

Abstract

Query Expansion was designed to overcome the barren query words issued by user and has been applied in many commercial products. This treatment tries to expand query words to identify users' real requirement based on semantic computation. It may be critical to deal with the problem of information overloading and diminish the using threshold, however the modern retrieval systems usually lack user modeling and are not adaptive to individual users, resulting in inherently non-optimal retrieval performance. In this study, we propose the LLSF method based on each individual search history to automatically generate specific personalized profile matrix. By which to generate context-based expanded query words. Considering the accuracy of retrieving performance, we process query words re-weighting algorithm to achieve this goal. Finally, the documents list is ranked by the way of stressed density distribution modeling. And the experimental result shows that our framework corresponds to personalization and the performance is very promising.

Keywords: Personalization, Latent Semantic Indexing, Query Expansion, Relevance Feedback, Maximum Entropy Density Function.

1. Introduction

1.1 Research Background and Motivation

The widespread usage of search engines has grown for many years, The searching technique can be used to apply in various aspects, either in World Wide Web or in particular Information Retrieval (IR) database. People can find what they want through it in the world of information overloading. For all of these reasons, search engine now becomes indispensable in the modern lives.

Traditional search engine presents search result based on keyword matching of users' query. It is the simplest method to gather documents associated with specific keywords. However, it's possible for users to acquire undesired results due to inadequate acquisition.. One of these problems is most common users of IR systems type short queries. (Shen et al., 2005) From a *single query, however, the retrieval system can only have very limited clue about the user's information need.*

Mostly, it's hard for users to realize what actual searching requirement is when proceeding searching activities, so there are certain of measure to solve the problem about vague queries issued from users in the past years.

1.2 Research Objective

In this article, we try to provide users more selective query keywords which are related to original query not only for the suggestion but also to help users realizing the real requirement in searching behaviors when users submit too brief query to find out more wanted documents. In addition, we also decide to deliver the decision making authority to users of which documents seem to be more preferred by them for achieving user center approach. Otherwise, in order to provide each user with more personal searching environment and contents, we endeavor to propose several approaches to adapt search results according to each user's information need.

2. Related Work

2.1 Review of Refining Short Query

To solve the problem of low retrieval performance caused by inappropriate query terms, automatic query expansion techniques have been studied for the past 30 years. In a recent study (Jansen et al., 2000, March 1), the number of query terms used by most end users was no more than 2 when searching with a Web search engine, which is even less than that of searching online databases. The same study also pointed out that only 5% of queries were accompanied by any relevance feedback feature.

2.2 Categories of Query Expansion

Query expansion techniques fall into two categories according to the way of implementation. One is to add new terms to an original query before searching, and the other is to formulate a new query on the basis of some retrieved documents of the previous search (Qiu, 1995). While the former is usually called a global or corpus-specific query expansion, the latter is called a local or query-specific query expansion. Global query expansion rely on thesauri that is a manually-built resource, as though WordNet-based (Mandala et al., 1999) provides the relation types include coordination, synonyms, hyponyms and etc for expanding the feature of original query terms.

Local query expansion, which corresponds to feedback retrieval, can acquire relevance information by either user feedback (Robertson & Sparck Jones, 1976; Rocchio, 1971) or system feedback. Query expansion using user feedback based on relevance judgment made by users, brought a significant improvement in retrieval performance (Harman, 1992 June; G. Salton & Buckley, 1990). Another two theories about local query expansion is Local co-occurrence method (HE et al., 2002; ZHANG et al., 2002) and Latent Semantic Indexing (LSI-based) (Deerwester et al., 1990)

2.3 Personal Data Construction

There are several way to gather the user's information for constructing unique data each end user belongs to. One of these approach is to have users describe their general interests. For example, Google Personal asks users to build a profile of themselves by selecting categories of interests. Google's PageRank algorithm can be described as personal web search techniques augmenting traditional text matching with a global notion of "importance" based on the linkage structure of the web. This global notion of importance can be specialized to create personalized views of importance.

User profile data provide information about the users of a Web site. A user profile contains

demographic information (such as name, age, country, marital status, education, interests, etc.) for each user of a Web site, as well as information about the users' interests and preferences. Such information is acquired through registration forms or questionnaires, or can be inferred by analyzing Web usage logs (Eirinaki & Vazirgiannis, 2003, February). Personal profiles can also be combined with the method mentioned above in the context of the Web search to create a personalized version of PageRank for setting the query-independent priors on Web pages. (Teevan et al., 2005). (Liu et al., 2002) used a similar technique for mapping user's queries to categories based on the user's search history.

3. Interactive IR system

Generally in interactive situation, system collects user's intention through designed interactive interface. In principle, every action of the user can potentially provide new evidence to help the system to better infer the user's information need. Thus in order to respond optimally, the system should use all the evidence collected so far about the user. After collection of the user information, how to effectively select and analyze these data is critical to this kind of system.

To retrieve more user demanded results, we carry out Linear Least Squares Fit (LLSF) algorithm to generate personal profile by matrix combination in which the personal searching result will be formed in document-term (DT) matrix, and Singular Value Decomposition (SVD) is used to reduce the dimensions of the original DT matrix. Moreover combining of document-cluster matrix and decomposed DT is to produce the final user profile M matrix. This process is also called Latent Semantic Indexing, which could extract the context-based terms out for expanding personalized query terms.

Simultaneously, as far as possible to promote the retrieval accuracy, relevance feedback of probabilistic model is suitable to be involved in. And in the traditional Retrieval method likes TF*IDF weighting schema, existing problem of mis-weighting could be caused the poor retrieval result. To overcome this defect, we adopt smoothing function of TF*IDF which could be diminished the inadequate weighting result. Finally we try to optimize the result representation, the ranking algorithm is also seen to be critical. For improving the Term Frequency (TF) ranking model, ranking function considering density distribution is brought into our framework.

3.1 Retrieval Method

After word recognition, each document is represented as a bag of words, but it does not mean that every word is a meaningful unit. For subsequently retrieval purpose, we need to set every recognized term a appropriate weight.

Furthermore, when users try to issue single or shorter query for searching, we use traditional keyword matching method to catch documents indicated by the user as relevant and conduct query expansion from these first time extraction documents in which we expect to get a list of longer query words, then we carry out traditional vector space model (VSM) to extract more query-relevant documents for generating more user demanded queries.

Because of the classical term weight model, TF*IDF scheme, usually has mis-weighting problem.

For example, a single document that contains the word “ERP” which only appears one time should not be deemed as relevant to a query containing “ERP” as a longer article that contains 20 occurrences of the word “ERP”. On the other hand, we ought not to assume that the longer document is 20 times more relevant. For this reason we prefer a smoothed version of TF and IDF(Croft & Harper, 1979) as listed below:

A common term frequency (TF) expression is then modified:

$$TF = \frac{f(K+1)}{f+KL} \quad (1)$$

where L = the normalized length of document D. If the document is of average length, then L = 1.0. K = a constant, usually set between 1.0 and 2.0. f = specific term occurs in single document. The TF component is designed to increase in value quite modestly as f arises. For instance, if f, K and L are 1, then $TF = 1.0$. If f were 9, then $TF = 1.8$. We can properly avoid the mis-weighting problem of conventional TF through this kind of effort. Smoothing inverse document frequency (IDF) prevents division by zero in the case where a term does not occur in the document collection at all.

$$IDF_t = \log\left(\frac{N-n_t+0.5}{n_t+0.5}\right) \quad (2)$$

where N = the size of the collection, n_t = the number of documents containing a given term, t.

3.2 Ranking Result

As noted above, a Boolean search generally returns sets of documents that are unordered, or ordered by certain criteria unrelated to relevance, such as time or date.

Most Web search engines are based on a different technology that ranks search results based upon the frequency distribution, term frequency, of query terms in the document collection. To cite an instance, if a document contains many occurrences of a query term “ERP”, this suggests that the document might be highly relevant to a query like “There are many software providers have ERP solutions, and the follow name lists which is one of the ERP providers?”

For this reason, we consider several criteria to consider document ranking score, then we expect document which is more relevant to user’s demand will be rank in higher place through sorting specific ranking score. The viewpoint of our criteria separated into four factors between single keyword and individual document. There respectively are similarity, density, term frequency and title appearance.

3.2.1 Similarity

We retrieve documents in VSM-model by comparing similarity information $sim(k,d)$ among a keyword k and a document d, then defining a positive threshold value for judging which one passing this value is seen to be relevant. So the single item gets a higher similarity value that we have confidence which one is more relevant to issued query keyword.

3.2.2 Maximum Entropy Density Function

By contrast, conventional ranking technology gives score to documents merely considered term frequency and regardless of the density distribution of specific keyword in subject document. But if

terms stated to be highly concentrated, it maybe mean that some topic is intensely described somewhere. So we carry out Maximum Entropy Function used to examine the density distribution of query keyword k, instead of just term frequency in considering document score. The original equation as formula (3) below, the value of E(K) becomes higher when p(k) in a average value that means probability distribution of k is more steady; E(K) has a lower value when p(k) is extremely in high and low value.

$$E(K) = -\sum_{k \in K} p(k) \log p(k) \quad (3)$$

So the entropy equation is revised to formula (4) (K. F. Jea & P. Y. Hsu, 2000), for ensuring the state between E(K) and p(k) is positive in synchronous up and down.

$$E(K) = -\sum_{k \in K} p(k) \log[1 - p(k)] \quad (4)$$

In physics, the meaning of density is that the degree of object distribution in the unit space. Accordingly considering the keyword density distribution in unit length of document will be more closed to reality and achieve the normalization.

After normalization adjustment, entropy equation is represented as follows formula (5):

$$E_s(K) = -\sum_{i=1}^n \frac{p_i(k) \log[1 - p_i(k)]}{S_i} \quad (5)$$

where $P_i(x)$ = the occurrence probability of term k in sentence i, S_i = the length of sentence i, n = number of sentences in a document. By this treatment, we can differentiate when document with same term frequency of query keyword, and then rank them by density distribution consideration.

3.2.3 Term Frequency

Although term frequency (tf) is basis to rank the documents, high occurrence of keywords in a document indicates that the weight of this document is remarkable significance. Therefore, we also adopt concept of term frequency to ensure our ranking model. But basic tf weighting method emerges the problem of mis-weighting, likes mentioned before, so we transfer raw $tf(k_i, d_j)$ into normalization according to maximum frequency of any term $Maxtf_j$ in a document d_j .

3.2.4 Title Appearance

When author composes particular topic, title often brings out overall theme or subject within article content. People surf on a search engine or even read news article, using title to decide whether to enter a website or further read an article they are interested is always an obviously evidence that these titles engage their concern. In the other words, if the numbers of query keywords k in a document's title t have a higher frequency $f(k_i, t_j)$ means that this article is considered to be more relevant by the user. We formulate an equation of this concept as $W_T * f(k_i, t_j)$, where W_T is a constant we can adjust to determine the weighted stress of this factor.

3.2.5 Rscore Ranking

To sum up these ranking factors, we merge these variables into single equation as formula (6):

$$R_{score} = F(k_i, t_j) * [sim(k_i, d_j) + E_s(K) + \frac{tf(k_i, d_j)}{Maxtf_j}] \quad (6)$$

$$F(k_i, t_j) = \begin{cases} 1, & f(k_i, t_j) = 0 \\ W_T * f(k_i, t_j), & f(k_i, t_j) \geq 1, \dots, n \end{cases}$$

where $f(k_i, t_j) \in N$, $W_T \in N$, W_T = weighting stress for occurrence of keyword in a title, $f(k_i, t_j)$ = occurrence of keyword k in a title t, $sim(k_i, d_j)$ = similarity value between keywords k and a document d, $E_s(K)$ = sum of each keyword's entropy value in a document d, $tf(k_i, d_j)$ = the frequency of keyword k in a document d, $Maxtf_j$ = the maximum frequency of any keyword k in a document d.

3.3 Query Expansion

3.3.1 Probabilistic Models of Query Expansion

In a probabilistic framework, selecting terms and computing relevance weights are treated as two different problems. This model is used to compute more accurate weight estimates. Consider the term incidence contingency table in Table 1.

Table 1. Term Incidence Contingency Table (Jackson & Moulinier, 2002)

	Relevant	Non-relevant	Total
Containing the term	r	n-r	n
Not containing the term	R-r	(N-n)-(R-r)	N-n
Total	R	N-R	N

where N = the number of documents in the collection, R = the number of relevant documents for this query, n = the number of documents having term t, r = the number of relevant documents containing the term t. The term weight from the equation which we mentioned above, would then be modified to take account of the relevance information as follows:

$$w_{t,d}' = \frac{f(K+1)}{f+KL} \log \frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \quad (7)$$

We utilize this re-expressed formula to re-weight the term within the vector space model when the user explicitly checks the retrieved document seen to be relevant or non-relevant. Subsequently, here address how terms are selected for expanding activity. In table 2, we can obviously observe the post weighted scores are risen when choosing a list of documents relevant to the topic of "ERP", "PeopleSoft", "SAP" and "Oracle" and non-relevant to "J.D.Edwards", "鼎新", "軟體部" and "Siebel".

Table 2. Comparison of Re-weight activity

	ERP	PeopleSoft	SAP	Oracle	J.D.Edwards	鼎新	軟體部	Siebel
Initial Weight	2.718	3.365	2.792	2.681	5.953	5.302	4.117	4.013
Re-weight	3.714	5.329	4.752	3.253	5.986	4.192	3.390	3.886

This model discussed by Robertson(1990) considers the distribution of scores for relevant and non-relevant documents. The model leads to an "offer weight", the larger the offer weight, the better the candidate, which is used to rank candidate terms.

$$OW_i = r_i \log \frac{(r_i + 0.5)(N - n_i - R + r_i + 0.5)}{(R - r_i + 0.5)(n_i - r_i + 0.5)} \quad (8)$$

This two model proposed by Robertson tightly integrates query expansion using relevance feedback and probabilistic retrieval.

3.3.2 LLSF Models of Query Expansion

At the beginning of the second stage expansion, we prefer to take user profile that not only has benefit to provide extra information about personal search intention, but also greatly reduce the falsehood of retrieved result. Furthermore, we adopt algorithm with respect to noise reducing, the Linear Least Squares Fit (LLSF) method proposed by (Liu et al., 2002), to construct matrix as personal user profile.

In order to have one of the matrixes, we first need to introduce our cluster method with respect to Single-Pass Clustering and 2-way K-Nearest-Neighbors (KNN) of Topic Detection and Tracking (TDT).

3.3.2.1 TDT Clustering

1. Detecting New Cluster

A large number of clustering methods were studied in IR research. This section we adopt the TDT proposed by CMU (Tang et al., 1999). One of the two algorithms is Single-Pass Clustering (SPC) for clustering task and the other is 2-way K-Nearest-Neighbors (KNN) for automatic classification. Figure 1. demonstrates SPC flow chart.

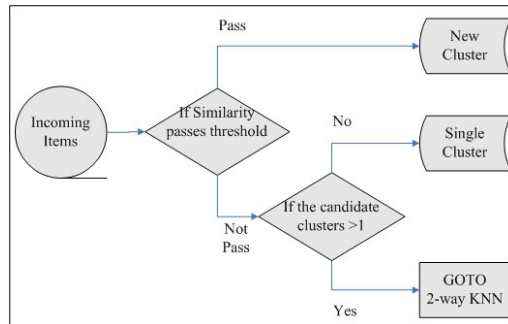


Figure 1. Flow chart of Single-Pass Clustering

Single-pass clustering follow the process as listed below and apply cosine as similarity calculation function: (1) Above all, taking out the first item in document collection as the first cluster. (2) Then take out the second item, calculating the similarity between item and clusters have been created. (3) If there is no similarity passes the threshold, instantaneously letting the incoming item be a new single cluster. (4) If the similarity passes the threshold we just set before, therefore categorize incoming item into appropriate candidate cluster. (5) If step 4 is selected, rescoring the centroid vector space of this cluster. (6) Iterating step 2 to 6, until dealing with entire incoming items.

2. Automatic Classifying

2-way KNN in TDT is used to classify the incoming item into proper classification by computing the relevance score. Which refers to compare objective cluster and else cluster that both take numbers of k Nearest-Neighbors. Objective clusters with respect to documents in this clusters which are prepared for

comparison; else cluster means documents in the clusters which different from objective clusters in the candidate clusters. Formula (9) explains the calculation of relevance score.

$$relevance_score(x, kp, kn, D) = \frac{1}{|U_{kp}|} \sum_{y \in U_{kp}} \cos(x, y) - \frac{1}{|V_{kn}|} \sum_{z \in V_{kn}} \cos(x, z) \quad (9)$$

By picking up the maximum relevance score which specific cluster belongs to, we can estimate this one is suitable to chosen for incoming item.

3.3.2.2 Algorithm to Learn Profile

1. Singular Value Decomposition

With regard to the meaning of SVD, we discuss it as follows. If there is a high dimension data, it can be applied SVD for dimension diminishing. In the linear algebra, SVD has a special characteristic to transform a high dimensional data to lower one. This method is often called matrix decomposition. By this way, high dimension matrix could be reduce to lower one then even achieve rule and noise reduction via selection singular value in diagonal matrix. The potential power of SVD is which can attempt to estimate the hidden structure and discover the most important associative patterns between words and concepts. Figure 2. demonstrates the process of the SVD:

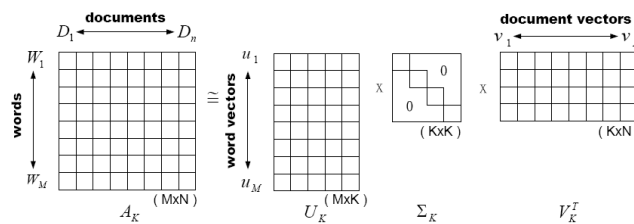


Figure 2. Singular Value Decomposition (SVD)

2. Rank selection

In re-composition process there is a critical point has to be taken notice. The diagonal matrix, we have to select precise rank k for diminishing the noises effectively. And how many rank k we should decide? One of these methods is to observe the singular value when they felled down from violent to smooth, and the previous of the margin value is the best choice. e.g. As the dotted line in the diagram below, there is a margin value k=9 for rank adoption.

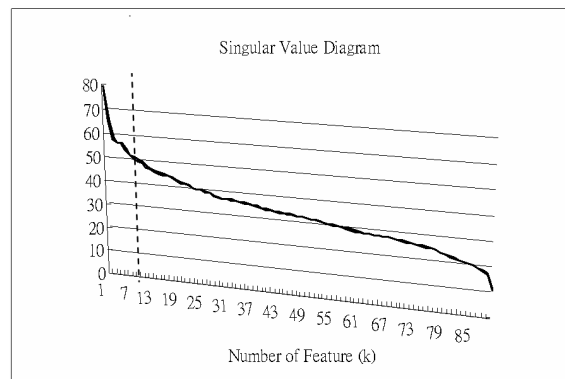


Figure 3. Singular Value Variation Diagram

3.3.2.3 Constructing the User Profile in Matrix Feature

The learning equation mentioned above is concerned with concept of Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI) (Deerwester et al., 1990). LSI is a theory for extracting and representing the relationship of words in a large corpus of text by using the co-occurrence of words and a mathematics technique, Singular Value Decomposition (SVD). In addition, there has another statement declared is LSI which could overcome crucial defect happened in searching process. This method projects documents and words to a predefined space, finding out the latent relationship between terms and documents. Even can retrieve the relevant documents when the situation that searching keywords is not appeared.

Given the m-by-n document-term matrix DT and the m-by-p document-cluster matrix DC, the Linear Least Squares Fit method computes a p-by-n cluster-term matrix M. In this step, techniques solving the problem is to employ the concept of Latent Semantic Index (LSI) in which Singular Value Decomposition (SVD) is the mathematical measure to decompose the input matrix. By this measure,

DT is decomposed into the product of three matrixes $U_k * \Sigma_k * V_k^T$, where U_k and V_k are orthogonal matrices and Σ_k is a diagonal matrix. After such decomposition, we can straightforward to recompose and combine DC matrix for computing particular matrix M, $M = DC^T * U_k * \Sigma_k^+ * V_k^T$, where Σ_k^+ is the inverse of Σ_k . Figure 6. illustrates the process of learning profile M:

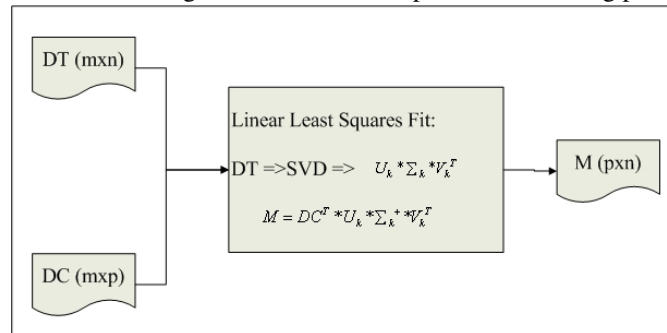


Figure 4. Process of learning profile M

We use matrices to represent the user's search histories, clusters of documents and user profiles as following Table 3.

Table 3. Document-Term matrix (DT)

Doc \ Term	.Net	Exchange Server 2003	Outlook	Palm	PocketPC	Palm OS
D1	5.0222	7.1262	3.3484	0	0	0
D2	0	11.7001	8.4147	0	0	0
D3	4.3711	10.4553	6.8124	0	0	0
D4	0	0	0	1.6325	0	0
D5	0	0	0	4.3454	6.5116	3.3619
D6	0	0	0	4.4678	5.6560	2.4723

Matrix DT (m*n) is a document-term matrix, m is the number of documents considered relevant by the user in a user's search history and n is the number of distinct terms occurring in these documents, which is established from user's query and the retrieved relevant documents the user indicates explicitly. The value of DT(i,j) is determined by the modified TF*IDF scheme.

Table 4. Document-Cluster matrix (DC)

Cluster Doc	Cluster 1037	Cluster 4194
D1	1	0
D2	1	0
D3	0	1
D4	0	1
D5	1	0
D6	0	1

Matrix DC ($m \times q$) is the document-cluster matrix, which is established from the relationships between the clusters and the documents. For each row in matrix DT, there is a corresponding row in the matrix DC. The columns of DC are the set of related clusters. If a row in DT represents a query/document, then the corresponding row in the matrix DC represents the set of clusters related to the query/document. Moreover, if there is an edge between the y -th cluster and the x -th query/document, then the entry $DC(x,y) = 1$; otherwise it is 0.

Table 5. Cluster-Term matrix M expresses a user profile

Term Cluster	.Net	Exchange Server 2003	Outlook	Palm	PocketPC	Palm OS
Cluster 314	1.3097	3.2141	2.0575	0	0	0
Cluster 184	0	0	0	1.3215	1.5242	0.7774

We have learned a matrix M ($p \times n$) from DT and DC, which is represented as the user personal profile. In this example, “Cluster 314” and “Cluster 184” are cluster field; “.Net”, “Exchange Server 2003” and “Outlook”...etc are term field.

3.3.2.4 From Profile to Expansion

Following the upper step, we have constructed personal profile in cluster-term matrix format from search’s history and latest relevant documents. Terms in the same cluster means that the relation among them are strongly recognized as Table 5., it can be used for expansion purpose when the one of the query keyword is appeared in this term list. This activity is described in Figure 5.

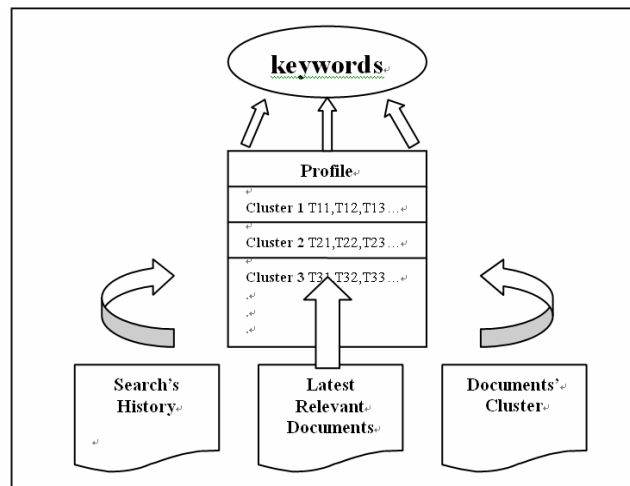


Figure 5. Concept of Profile Expansion

4. PNQES: A Personalized Search Engine

4.1 Design

To experiment with the personalized environment, we create the PNQES search engine. This personalized agent could provide the user a query expansion function which is separated into two stages. Above all, system will automatically catch and parse the query terms when the user has submitted completely. After parsing, search component with VSM-based search going to weight each words in the query according to vector space model (VSM) strategy for retrieving all possible documents related to the original query.

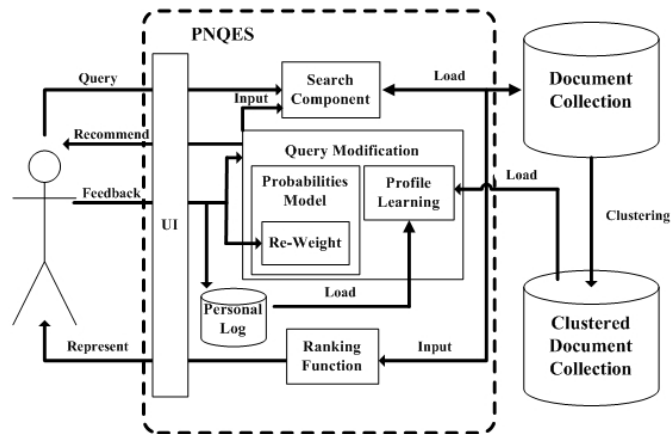


Figure 6. NQES Architecture

Next step, ranking component has considered of several factors which influence the ranking result with similarity, dense distribution, term frequency and title occurrence. Simultaneously this ranked result would be stored in personal log file for later analyzed.

For finding potential search's intention, system will ask the user to respond some feedback, called "Relevance Feedback", with judgment whether documents set is relevant or not while showing the retrieved result on the screen. In the meanwhile, when a user submits this response, Query Modification component adopted probabilistic model is able to give all terms in all relevant items with "Offer Weight" and then output some candidate terms in first stage expansion. In the parallel step, LLSF component combines personal search's history and pre-clustered corpus applied TDT algorithm to construct matrix called personal user profile in second stage query expansion.

Lastly, query issued by the user will be expanded to a number of proper personal keywords via this two stage expansion processes. Figure 6. is illustrated PNQES system architecture.

4.2 Experimental Data Sets

The experiment target where we focus on is enterprise technology reports because most terms contained in are consisted of proper noun in which the experimental result can seem to be more accuracy. Moreover we try to collect data from Website, Taiwan.CNET.com, which contains various documents associated with specific software techniques and hardware information and the reported

date from January 4, 1999 to April 31, 2006.

This corpus has been separated to two main classes, Enterprise Application and News, and the 9 sub-topic. The volumes distribution of these topics is shown in Table 6.

Table 6. Volumes of Corpus

Sub-Topic	Category	Numbers of Article
IT techniques	EA	1000
Special Topic Report	EA	250
Case Study	EA	550
Special Column	EA	1137
Research Report	EA	1511
Enterprise Software	News	6660
Enterprise Hardware	News	4991
Network/Communication	News	2796
3C Product	News	3694
Total Volumes		22988

p.s: EA: Enterprise Application

4.3 Word Recognition

Word segmentation is crucial for the research of information retrieval, especially for Chinese documents. The reason is that there is no word boundary in sentences, which increases the difficulty of this work. In this research, we extract Words with respect to verbs and nouns in “Eighty Thousand dictionary” that is published by Institute of Information Science Academia Sinica, then merge them into another dictionary possessing names of location, Institute and company and gathered by our laboratory. Moreover, we extract terms from a document by principle that treats the long-term has a higher priority than others, when this step is over, next we apply the newest version of word segmentation system developed by Chinese Knowledge Processing Group (CKIP) to pick up the rest terms that Words database can not capture for ascending the precision of recognition.

5. Evaluation of PNQES

5.1 Evaluation Method

5.1.1 Experimental Subject

Because of our experimental corpus is focused on IT related articles, testers have be expected to hold the professional IT knowledge of how to realize which article topic is their demand one. For this reason, we plan to ask 10 users who both are the graduate students and major in the department related to “Computer Science”.

5.1.2 Evaluation Variable

The evaluation step symbols are described as : (1)*QTR*: Query Term Re-weighting, (2)*LLSF*: Linear Least Square Fit Expansion Model, (3)*Baseline*: We execute raw query expansion without query term re-weighting and Linear Least Square Fit Expansion Model analysis, then using NAP (Non-interpolated Average Precision Rate) to evaluate the precision values.

5.1.3 Evaluation Procedure

And then we design an evaluation procedure with regard to retrieval and ranking precision. The brief

evaluation process has listed in table 7.

Table 7. Brief Evaluation Process

Event Number	Evaluation Event
E_01	Precision in baseline event
E_02	Precision in baseline + QTR event
E_03	Average R-precision in TF ranking model
E_04	Average R-precision in Rscore ranking model
E_05	Precision LLSF model based on baseline

5.1.3.1 Description of Evaluation Processes

1. System Initiation:

To actually simulate the real condition, we request tester to input single query to initial the system. Firstly, system will retrieve the keyword related articles by “Boolean AND search” method, and all articles which contain this keyword will be retrieved and ranked in TF ranking approach.

2. Query Expansion:

While possibly related items have been retrieved, the testers will be asked to indicate several articles which they think to be relevant and are explicitly stored into personal log, furthermore the submitted query will be expand from analyzing these relevant items by probabilities model. The testers at will select a number of recommended keywords with scattering issues and then add them to original query list for the following search. The expanded query list will conduct VSM search for retrieval task.

3. Training and Ranking:

In ranking test, we ask tester to interact with our system for 3 time as step 2 for successful training the query list to robustness, so that the after training list is for doing the baseline task. Further, testers who evaluate the TF ranking and Rscore ranking also based on this baseline with the evaluation formulation of ranking function “Average R-precision”.

4. Methods Integration and Profile Recording:

The following retrieval tasks we increase one variable “QTR” to our evaluation activity with the ranking function Rscore measure. This step has two objectives, one is to evaluate the usability of variable QTR and the other is to as far as possible make increasing of interaction between tester and system for establishing personal search’s history completely.

5. Two Expansion Methods Comparison:

When tester searching behavior has been fully caught, we ask testers to evaluate the results expanded by LLSF model based on baseline to contrast the variation of two models that baseline means rarely using the probabilities model and the latter means a hybrid expanding activity with two expanding model.

5.1.4 Retrieval and Ranking Statistics

First, the evaluation functions we referred is TREC_EVAL method developed by Buckley(1991). We have altered to fit condition of our experiment. Table 8. is the example of one of our scored cards. The rest cards we have appended to appendix A.

Table 8. EVAL Scored Card A

User 1	
Queryid (Num) 1	
Precision for all relevant documents	
E_01	0.8333
E_02	0.9286
E_03	0.8846

EVAL Scored Card B

User 1	
Queryid (Num) 1	
Total number of documents over all queries	
Retrieved:	23
Rel_ret:	20
Precision: (in TF ranking)	
At 5 docs:	0.8000
At 10 docs:	0.8000
At 15 docs:	0.7333
At 20 docs:	0.7500
At 30 docs:	0.6667
Average R-Precision:	0.7500
Precision: (in Rscore ranking)	
At 5 docs:	1.0000
At 10 docs:	1.0000
At 15 docs:	0.9333
At 20 docs:	0.9000
At 30 docs:	0.6667
Average R-Precision:	0.9000

p.s: Rel_ret: Retrieved Relevant document

In the experiment, we have recorded retrieved result set and relevant items from evaluation process E_01 to E_05. After recorded, we turn the data to scored card format and draw the bar chart for observing if each effect variable has been added respectively, the result precision will be changed significantly. The evaluation method we have adopted the precision for result retrieval and average R-precision for result ranking. We also have compared whether or not the Rscore Function is significantly better than Term Frequency (TF) ranking method.

5.2 Experiment Result

1. Result of Each Retrieval Methods:

First of all, we examine the baseline and of combining the QTR as E_01 and E_02 to observe the variation of precision. The variation of each variable appended is demonstrated in Figure 19. We can see E_02 significantly outperform the baseline. It is clearly demonstrates that it is worthwhile to combine the QTR to yield higher retrieval precision.

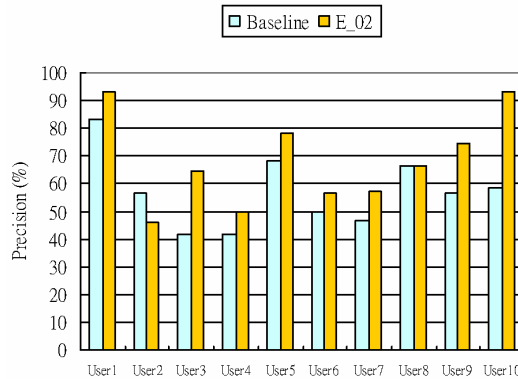


Figure 7. Precision of different combining methods to 10 users

2. Result of Two Expansion Models:

Then add UP to evaluate the personalized search as E_05. Another observation from Figure 8. is that using the UP to revise the expanding terms, this approach gives extraordinary precision value than rarely using classic probabilities model alone. This tends to imply that the personal profile is worthwhile to perform personalized search.

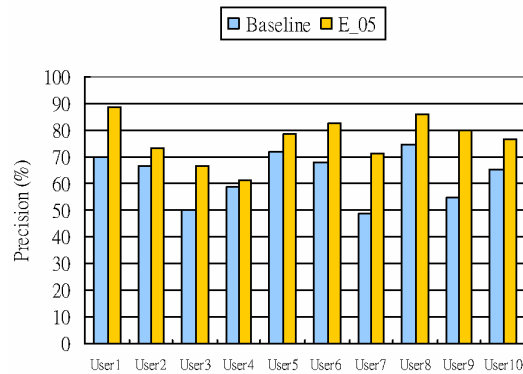


Figure 8. Results of adding the UP analysis

3. Result of Two Ranking Methods:

Distribution of Average R-Precision value presented in Figure 9., applying Rscore Function obviously performs a higher average precision than original TF ranking measure. So we firmly trust that consideration of several factor mentioned in section 3.2 when undertakes ranking task will induce the performance improvement.

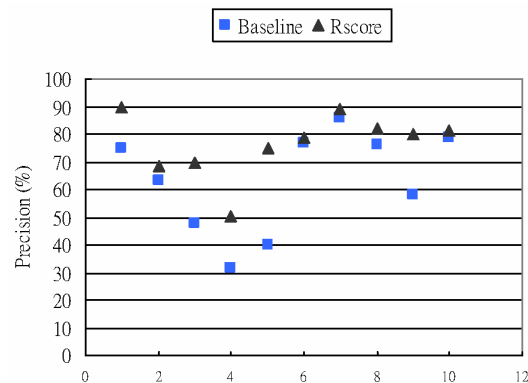


Figure 9. Average R-Precision with two ranking method

6. Conclusion

In this paper, we propose a mechanism which can be intelligent to learn the user's search behavior and provide specific search results for each differentiated end-users. To achieve this purpose, adopting 2 stages query expansion and hybrid density distribution ranking function is our efforts.

Query Expansion activity in first stage we have applied probabilities model which takes that the expanding and weight re-calculating as different parts, both are based on the relevant documents of user's feedback. While in stage 2 expansion, system initiatively combine the personal profile and latest relevant items indicated by the user and transform with respect to LLSF metrics merging procedure to extract out more suggested terms of user-driven's. As the list of documents have been retrieved

completely, so as to show the most relevant items for the user, ranking method we have considered several influence factor to give the appearance priority to each items.

Furthermore we utilize evaluation criteria to prove our PNQES is of feasible and effective. And the result performance has proved this proposed system framework not only could be applied in local database, but also could be well-performed in web-based searching for personalization enhancement.

7. Reference

1. Deerwester, Fumas, Landauer, & Harshman. (1990). *Indexing by intent semantic analysis*. Paper presented at the JASIS.
2. Eirinaki, M., & Vazirgiannis, M. (2003, February). *Web mining for web personalization*. Paper presented at the ACM Transactions on Internet Technology (TOIT).
3. Harman, D. (1992 June). *Relevance feedback revisited*. Paper presented at the Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, New York.
4. HE, H.-z., HE, P.-l., GAO, J.-f., & HUANG, C.-n. (2002). Query expansion based on the context in chinese information retrieval. *Journal of Chinese Infomation Processing*, 16(6), 32-37.
5. Jansen, B. J., Spink, A., & Saracevic, T. (2000, March 1). Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing & Management*, 36(2), 207-227.
6. Liu, F., Yu, C., & Meng, W. (2002, November 4--9). *Personalized web search by mapping user queries to categories*. Paper presented at the In Proceedings of CIKM, McLean, Virginia, USA.
7. Mandala, R., Tokunga, T., & Tanaka, H. (1999). *Combining mutiple evidence from different types of thesaurus for query expansion*. Paper presented at the SIGIR.
8. Qiu, Y. (1995). *Automatic query expansion based on a similarity thesaurus*. ETH Zurich.
9. Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, 27(3), 129 - 146.
10. Rocchio. (1971). Relevance feedback information retrieval. In *The smart retrieval system-experiments in automatic document processing* (pp. 313 - 323). Kansas: Prentice-Hall.
11. Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *41*(4), 288 - 297.
12. Shen, X., Tan, B., & Zhai, C. (2005, August 15-19). *Contextsensitive information retrieval using implicit feedback*. Paper presented at the SIGIR, Salvador, Brazil.
13. Teevan, J., Dumais, S. T., & Horvitz, E. (2005, August 15--19). *Personalizing search via automated analysis of interests and activities*. Paper presented at the SIGIR, Salvador, Brazil.
14. ZHANG, M., SONG, R., LIN, C., MA, S., JIANG, Z., LIU, Y., et al. (2002). *Expansion-based technologies in finding relevant and new information*. Paper presented at the TERC.