# 國　立　中　央　大　學

## 資 訊 管 理 研 究 所
## 博　士　論　文

# A study of relevance feedback on retrieved documents in a vector-space-modeled system


研 究 生：張維平

指導教授：周世傑 博士



中　華　民　國　九　十　六　年　六　月

# 國立中央大學圖書館
# 碩博士論文電子檔授權書

(95 年 7 月最新修正版)

本授權書所授權之論文全文電子檔(不包含紙本、詳備註 1 說明)，爲本人於國立中央大學，撰寫之碩/博士學位論文。(以下請擇一勾選)

(V)同意 (立即開放)
( )同意 (一年後開放)，原因是：______
( )同意 (二年後開放)，原因是：______
( )不同意，原因是：______

以非專屬、無償授權國立中央大學圖書館與國家圖書館，基於推動「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、微縮、光碟及其它各種方法將上列論文收錄、重製、公開陳列、與發行，或再授權他人以各種方法重製與利用，並得將數位化之上列論文與論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名: 張 維 平　　學號：89443009

論文名稱: A study of relevance feedback on retrieved documents in a vector-space-modeled system

指導教授姓名：周 世 傑 博士

系所 ：資訊管理研究所 ■博士班 □碩士班

日期：民國 96 年 7 月 11 日

備註：

1. 本授權書之授權範圍僅限電子檔，紙本論文部分依著作權法第 15 條第 3 款之規定，採推定原則即預設同意圖書館得公開上架閱覽，如您有申請專利或投稿等考量，不同意紙本上架陳列，須另行加填聲明書，詳細說明與紙本聲明書請至 http://blog.lib.ncu.edu.tw/plog/ 碩博士論文專區查閱下載。
2. 本授權書請填寫並**親筆**簽名後，裝訂於各紙本論文封面後之次頁（全文電子檔內之授權書簽名，可用電腦打字代替）。
3. 請加印一份單張之授權書，填寫並親筆簽名後，於辦理離校時交圖書館（以統一代轉寄給國家圖書館）。
4. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

# 國立中央大學博士班研究生

# 論文指導教授推薦書

資訊管理研究所 張維平 研究生

所提之論文 A study of relevance feedback on retrieved documents in a vector-space-modeled system

係由本人指導撰述，同意提付審查。

指導教授 周 世 傑 (簽章)

96 年 6 月 11 日

# 國立中央大學博士班研究生

# 論文口試委員審定書

<u>資訊管理</u> 學系/研究所 <u>張維平</u> 研究生所提之論文 <u>A study of relevance feedback on retrieved documents in a vector-space-modeled system</u> 經本委員會審議，認定符合博士資格標準。

學位考試委員會召集人 徐世輝

委員 范[illegible]安

鄭晉昌

[illegible]信[illegible]

周世傑

中華民國 96 年 6 月 25 日

# 向量空間模式系統中對於檢索文件相關回饋之研究

## 論文摘要

在向量空間模式資訊擷取系統上，相關回饋是一種應用於提升擷取效率的技術。相關回饋的技術係在檢索過程中，由使用者對於系統檢索出的文件，進行相關或不相關的評估。過去的研究，主要在運用使用者回饋的資訊，修改使用者的興趣向量。本研究找出，過去研究在使用者相關或不相關的回饋文件中，未完全被研究過的資訊。這些資訊是關於字詞在相關或不相關文件中，所出現的各種狀態。本研究發展一個實驗性的資訊擷取系統與方法，以展示對於字詞出現狀態資訊的應用，並進行相關實驗，以研究這些方法是否具有效果。本研究實驗的結果顯示，字詞出現狀態的資訊是可以被抽取出來，並可應用於提升擷取效率。

關鍵字：資訊擷取、相關回饋、詞頻、敏感度、詞語權重、全球資訊網

# A study of relevance feedback on retrieved documents in a vector-space-modeled system


## Abstract

Relevance feedback is one of the techniques applied in a vector-space-modeled Information Retrieval (IR) system to enhance retrieval effectiveness. The feedback process usually has the user rate the documents retrieved as relevant or non-relevant. Most past studies apply the information of document relevance to the modification of the vector that is used to manifest the user's information interest. In this study, we have identified additional information obtained from relevance feedback that was not fully studied in the past from the rated relevant/non-relevant documents for application. The information pertains to is about the situations of term appearances in the relevant/non-relevant documents. We have developed a method together with an IR system to demonstrate the application of the information of term appearance situation. Experiments have also been conducted to study its effect. The experimental results preliminarily show that the information of the term appearance situation could be extracted and appropriately applied to enhance retrieval effectiveness.



Keywords: Information Retrieval, Relevance Feedback, Term Frequency, Sensitivity, Term Reweighting, World Wide Web.

## Acknowledgements

There are many people to whom I owe a debt of thanks for their support over the last seven years. First, I would like to sincerely offer my deepest appreciation to the members of my thesis committee; Dr. Fan, Dr. Jehng, Dr. Kao, Dr. Sheu and especially Dr. Chou whose boundless patience, continued instruction, encouragement, and expert guidance made this project a most rewarding learning experience.

I would also like to thank Dr. Hinchun Chen for providing me the opportunity to conduct short-term studies at the University of Arizona in 2002, and to Dan and Fiona Grady for providing tremendous help to my family during our stay there.

To my friend, Mr. Eric Lee, the Director of High Technology Crime Prevention Center, Criminal Investigation Bureau, Taiwan, I offer my thanks for his encouragement and gentle persuasion towards finishing my thesis.

Finally, my special appreciation is extended to my wife, Janet, for accompanying me on this most enlightening journey.

Taoyuan, Taiwan, June 2007

Weiping Chang

# Index

# Figure index

# Table index

# 1. Introduction

Due to the explosive growth of information on the Internet, contemporary Internet users may face the situation of information-overload. Studies on vector-space-modeled information retrieval (IR) systems have applied many techniques to limit the amount and increase the relevance of information retrieved. Relevance feedback is one of the techniques applied. Applications of relevance feedback in vector-space-modeled IR system usually have the user rate the retrieved documents as relevant or non-relevant, and then extract information from the rated documents for use. In most past studies, the information extracted for application consists of two sets of terms with frequencies: (1) terms appeared in relevant documents; (2) terms appeared in non-relevant documents. The two sets of terms can form two vectors and a vector merging operation based on addition and subtraction according to term relevance is used to expand query.

In this study, we have identified additional information from among the rated documents. To the best of our knowledge, the information that we have identified was not fully studied in the past and could be valuable to be extracted and applied in the enhancement of information retrieval. Consider the following three situations of term appearances in the relevant/non-relevant documents:

(1) a term can appear in relevant documents only and never appear in non-relevant documents;
(2) a term can appear in non-relevant documents only and never appear in relevant documents;
(3) a term can appear both in relevant and non-relevant documents.

Our interest here is that could the information of "term appearance situations" (abbreviated as *tas* later) just mentioned be extracted and effectively applied to enhance information retrieval. Considering that terms with different *tas* could be of different usefulness and importance in the manifestation of the user's interest and disinterest, the following two statements are worthy of study.

Terms belonging to *tas* 1 and 3 could have different power in the expression of the user's interest.

Terms belonging to *tas* 2 could be of great importance in the expression of the user's disinterest.

Therefore, the primary purpose of this research is, first, to develop a method together with an IR system to demonstrate the application of the terms belonging to *tas* 1 and 3 in the expression of the user's interest and the application of the terms belonging to *tas* 2 in the expression of the user's disinterest and, second, to study the effect of the method of application on the enhancement of information retrieval.

In Section 2, we will review some relevance feedback foundations and studies on the application of the information extracted from rated documents to provide the research basis to understand what else could be developed to complement the present findings. We will also propose the method of the application of the information of *tas* and finally initiate the design for an IR system. In Section 3, we will develop a vector-space-modeled IR system and have the embodiment and details of the design initiated in Section 2 embedded. In Section 4, we conduct some experiments with the IR system that we have developed to study the effect of the information extraction and application. Finally, we make some conclusions and offer some recommendations for future research in Section 5.

## 2. Relevance Feedback

Relevance feedback has been applied in many fields. It was first designed and used in vector-space-modeled IR system by Rocchio in 1966 as a method to increase the number of relevance documents retrieved by a query [27]. Later, it had been applied in other IR models. For example, Robertson and Sparck Jones proposed a relevance feedback method for a probabilistic retrieval system [26]; Dillon and Desper experimented with relevance feedback on a Boolean model [9]. Besides IR system, techniques like Neural Networks [7], Genetic Algorithms [3] and Machine Learning [10] also have relevance feedback applied.

The study of relevance feedback in this research focuses on a vector-space-modeled IR system. Usually, the manipulation on relevance feedback in vector-space-modeled IR system requires the user to rate the relevance of an initial sample of documents retrieved and the IR system is designed to have the valuable information residing in the user's relevance rating extracted and applied to enhance the effectiveness of information retrieval. Our interest here focuses on the extraction and application of the valuable information residing in the rated relevant/non-relevant documents. As aforementioned, we have identified some other information residing in the rated relevant/non-relevant documents that was not fully studied in the past and that could be valuable to the enhancement of information retrieval. In the following, we review some past studies about the extraction and application of the information residing in the rated relevant/non-relevant documents as the basis where our study tries to provide some complementary work.

The research of relevance feedback is divided into 3 stages, 1960's – 1970's, 1980's – 1990's, and 2000's, as follows, and many researchers followed or modified Rocchio's study to develop their IR systems. They will be introduced according to time sequence.

(1)1960's – 1970's: the initial stage

From 1960 to 1979, relevance feedback technique was initialized at this stage. The representative researchers of this stage were Rocchio and Ide. In 1966, Rocchio conducted the initial and widely known study on using relevant and non-relevant information from user's feedback to improve query performance. The principle of Rocchio's study was to adjust the query vector according to the information from user's relevant and non-relevant feedback. In the vector space model, each document and query are thought of as an n-dimensional vector space, where each dimension represents an index term with a weight. A vector merging operation based on addition and subtraction can then be used to expand queries by adding all the terms that are in the retrieved documents and then weighting terms are assigned according to document relevance. Rocchio's original formula is shown as follows:

$$Q_1 = Q_0 + \beta \sum_{k=1}^{n1} \frac{R_k}{n_1} - \gamma \sum_{k=1}^{n2} \frac{S_k}{n_2}$$

Where

$Q_1$: new query vector

$Q_0$: initial query vector

$R_k$: vector for relevant document $k$

$S_k$: vector for non-relevant document $k$

$n_1$: number of relevant documents

$n_2$: number of non-relevant documents

$\beta$ and $\gamma$: weight multipliers to control relative contributions of relevant and non-relevant documents.

In the formula, the term was re-weighted by adding the weights from the actual appearance of those query terms in the relevant documents, and subtracting the weights of those terms appearing in the non-relevant documents.

In 1971, Ide had developed two different feedback strategies, Ide Regular and Ide dec-hi, as follows [17].

Ide Regular $Q_1 = Q_0 + \sum_{k=1}^{n1} R_k - \sum_{k=1}^{n2} S_k$

Ide dec-hi $Q_1 = Q_0 + \sum_{k=1}^{n_1} R_k - T$

Where

$Q_1$: new query vector

$Q_0$: initial query vector

$R_k$: vector for relevant document $k$

$S_k$: vector for non-relevant document $k$

$T$: the top non-relevant document

$n_1$: number of relevant documents

$n_2$: number of non-relevant documents

The basic operational procedure of Ide Regular and Ide dec-hi was the merging of document vectors and original query vectors. Like Rocchio's original formula, Ide's two methods reweighted query terms by adding the weights from the occurrence of those query terms in the relevant documents, and subtracting the weights of those terms occurring in the non-relevant documents. Queries were expanded by adding all the terms not in the original query that were in the relevant documents and non-relevant documents. They were expanded using positive and negative weights based on whether the terms came from relevant or non-relevant documents. Different from Rocchio's original formula, the Ide dec-hi method only used the top non-relevant document for feedback, instead of all non-relevant documents retrieved within the first set shown the user.

(2)1980's – 1990's: the developing stage

From 1980 to 1999, some researchers applied relevance feedback to develop their IR systems. A summary of this research appears below.

In 1983, Salton et al. applied relevance feedback with query expansion on an extended Boolean IR model. In Salton et al.'s study, relevance feedback was applied into a Boolean IR model. To generate improved query statements, Salton et al. used automatic feedback techniques for Boolean query statements in online information retrieval based on information contained in previously retrieved documents [29].

In 1992, Harman experimented with the effect of relevance feedback on an IR system and applied an effective feedback technique in his research. The effective feedback technique used in Harman's study, termed *dec hi* used all documents in the positive or relevant feedback set and subtracted from the query only the vectors of the highest ranked non-relevant documents in the negative or non-relevant feedback set. The experimental result showed that query expansion and query reweighing were important to Harman's IR system, and the most improvement was from query expansion. Additionally, adding some amount of well-selected terms could improve the performance [13].

In 1995, Buckley et al. used a weighting scheme based on Rocchio's approach to develop their work on Dynamic Feedback Optimization [4].

In 1997, Singhal et al. applied Rocchio's algorithm to implement learning routing queries in a query zone in vector space model [32]. Also in 1997, Balabanovic et al. proposed an updated rule different from Rocchio's original formula for relevance feedback [2] as follows.

$$u(w,m,s) = m + sw$$

Where

$u(w,m,s)$: a function returning an updated user profile $m$ given the user's feedback $s$ on a page $w$.

$w$: a representation of a web page.

$m$: a representation of the user's interests.

$s$: the user's score for web page $w$ (3, 2, 1, 0, -1, -2, -3)

In 1999, Ng et al. combined the use of Rocchio's formula for term selection to create a hybrid algorithm for the routing task [24].

(4)2000's – the integrated application stage

From 2000 to present, relevance feedback technique is studied and processed in different approaches and some techniques are combined with relevance feedback to develop integrated IR systems. Prior to this period, relevance feedback was already a mature technique. It is widely applied into many domains, such as music, medical, etc. At last, relevance feedback used on WWW will be introduced.

In 2000, Hoashi et al. applied Rocchio's algorithm to develop a filtering system [14]. Also in 2000, Desjardins et al. developed IntellAgent to optimize the user profile. The algorithm of IntellAgent was a combination of the relevance feedback process and a genetic algorithm. When IntellAgent proposed a document, the user evaluated its relevance and replied "1" if it was relevant or "-1" if it was non-relevant. IntellAgent used this information to modify the weights of the firing vectors in the profile. The weights were modified according to the following formula [8]:

$$w_{ik}^{p} = w_{ik}^{p} + \alpha \times f \times w_{k}^{d}$$

Where

$\alpha$ : the feedback power a predetermined parameter between 0 and 1,

$w^{p}$ : the weights of the firing vectors of the profile,

$w^{d}$ : the weights of the proposed document

$f$ : the user feedback.

In 2001, Kim et al. reweighed the terms by adding their relevance degrees to their initial weights on a vector space model IR system. The relevance degree was calculated by fuzzy inference using the information such as co-occurrence similarity, document frequency within the feedback documents and the inverse document frequency [19]. Also in 2001, Nick and Themis developed Webnaut learning agent to collect the user's rankings on retrieved documents and altered the frequencies of words according to the following update rule [23]:

$$Tf_D = Tf_D + \left( \frac{c}{100} \times Tf_R \right)$$

Where

$Tf_D$ : the word frequency in the Dictionary,

$Tf_R$ : the word frequency in the recommended URL,

$c$ : the user's evaluation in the range -2 to 2.

In 2004, Savoy adopted Rocchio's approach into Effective European Monolingual Information Retrieval System [30]. Also in 2004, Moyotl et al. used Rocchio's method to develop a text categorization system [21]. Still in 2004, Azimi-Sadjadi et al.'s modified Rocchio approach to propose a retrieval system for a large database and for a large number of most commonly used single-term or multi-terms queries. In Azimi-Sadjadi et al.'s proposed system, vector space modeling was used to represent the attributes (terms) of the document and the proposed system consisted of a three-layer linear network with connection weights that corresponded to the tokens and their importance in documents in the original training database. A centroid learning method was presented to find an optimal query in the space spanned by the documents. Azimi-Sadjadi et al.'s retrieval system was capable of continuously learning from multiple expert users using a class of relevance feedback learning [1].

In this stage, some researchers combined other techniques with relevance feedback to develop integrated IR systems. In 2000, Crestani combined neural networks with relevance feedback in his study. The results of Crestani's study showed that the combination of the two techniques is more effective than both techniques taken separately [7].

In 2001, Drucker et al. applied Support Vector Machines into relevance feedback on their IR system. Drucker et al. found Support Vector Machines had very good performance when the amount of the documents returned was low and the number of relevant documents was small [10].

Relevance feedback studied and developed very well in this stage, hence it is applied in many domains, such as medical, music, etc., as follows.

In 2002, Hoashi et al. applied relevance feedback to develop content-based music IR system. In their music IR system, Hoashi et al. used feedback techniques to improve the music retrieval performance and the effectiveness of their IR system was obtained [14 ].

In 2006, Graugaard proposed methods for correlating a performer and a synthetic accompaniment based on Implicit Relevance Feedback (IRF) using Graugaard's expanded model for interactive music [12].

In 2007, Rho et al. developed a music IR system that incorporated user relevance feedback with genetic algorithm to improve retrieval performance. The experimental results showed that Rho et al.’s IR system had good effectiveness and efficiency [25].

With regards to applying relevance feedback in medical domain, relevance feedback is used for spine X-ray retrieval system.

In 2004, Shin et al. applied a query expansion strategy through automatic relevance feedback to search MEDLINE, a very large database of abstracts of research papers in medical domain, maintained by the National Library of Medicine. Shin et al.’s approach obtained improvement of the retrieval quality in MEDLINE [31].

In 2005, Xu et al. applied relevance feedback to develop a spine X-ray retrieval system. Xu et al. proposed a novel linear weight-updating approach for relevance feedback applying to spine X-ray image retrieval. The result of Xu et al.’s study indicated that the proposed approach could extensively enhance the retrieval performance to better satisfy the individual user's preferences [35].

In 2006, Christiansen et al. applied relevance feedback to refine query for PDF medical journal articles. In Christiansen et al.’s study, they used relevance feedback as an alternative to keyword-based search engines for sifting through large PDF document collections and extracting the most relevant documents [6].

Also in 2006, Wei et al. proposed an approach to learn pathological characteristics from user’s relevance feedback for content-based mammogram retrieval. Wei et al.’s experimental results showed that their approach effectively improved the average precision rate through five iterations of relevance feedback rounds [34].

Regarding relevance feedback applied on World Wide Web, relevance feedback used in the web search engine allows for analyses to be performed nearly in the same way as is conventional in IR systems. Relevance feedback is applied in many web based search engines, for instance Excite (http://www.excite.com) and Lycos (http://lycos.com), where the system presents a choice of words to the user and allows the user to expand the query based on those words. Moreover, applying relevance feedback to web search engines requires document representations to be descriptive. Indexing the entire document is a way to properly represent the document. Several researchers studied web information retrieval using relevance feedback. For instance, Hoeber et al. developed a web search system which allowed the user to interactively re-sort the search results based on the frequencies of the selected terms within the document surrogates, as well as to add remove terms from the query, generating a new set of search results [16]. Li et al. proposed an approach towards intelligent information retrieval by providing clustered web pages and mined concepts based on results of search engines [20]. Navigli et al. developed a web IR system. Both expanded the query using thesauruses and they showed that this proposal improves web information retrieval [22].

Studies cited above basically have the following information extracted and applied:

1. terms and frequency belonging to relevant documents;
2. terms and frequency belonging to non-relevant documents;
3. number of relevant documents;
4. number of non-relevant documents;
5. the user's ranking score.

In this study, we have identified some other information "*tas*" as aforementioned. Based on the ideas of past studies that have achieved successful performance, our propositions for the application of *tas* are as follows:

(1) The terms belonging to *tas* 1 and 3 could be used with different weight to show their

relative importance (termed as ‘sensitivity’ in the paper) in the vector to express the user’s interest. In the application of the vector, these terms could be used to provide query string key words with different priority and used as the basis with different importance in the similarity comparison to other vectors.

(2) The terms belonging to *tas* 2 could be used in the vector to express the user’s disinterest. It could be used to provide key words in the query string with “NOT” operator.

The information we have identified still resides in the rated relevant/non-relevant documents. The application of it would not deviate too far from Rocchio and the related studies. However, it could provide a different way of consideration and additional support to the enhancement of information retrieval.

Therefore, the extraction and the application of *tas* could be located as complementary. In this study, we experiment with the effect of the propositions mentioned. According to the propositions, the design of the system for the expression of the user’s information requirement can be specified as follows:

- Maintain a positive user profile to contain terms appeared in relevant documents and a negative user profile to contain terms appeared in non-relevant documents to support the extraction of term appearance situations.
- Maintain a positive user profile weighted by term’s frequency and term’s relative importance together to express the user’s interest and use this expression to provide key words for the generation of query string and basis for similarity comparison between the user’s interest and the retrieved document.
- Maintain a negative user profile to contain terms appeared in non-relevant document only and weighted by term’s frequency to express the user’s disinterest and use this expression to provide key words for the generation of query string with “NOT” operator.

# 3. The Experimental Vector-Space-Modeled System

Based on the system design initiated in the previous section, we have developed an IR system, EIRS (Experimental Informational Retrieval System), to extract and apply the information of *tas*.

In this section, the system framework of EIRS is introduced first, followed by a review of the system flows.

## 3.1 System framework

Figure 1 shows the system framework of EIRS. It contains five main modules denoted by the solid line rectangles. Two modules denoted by the dotted line rectangles outside the basic system are used to support the experimental process. The functional summary of each module is described as follows.

**User Input and Feedback**: This module supports the input of one example document originally, the input of the retrieved documents and the user's rating of relevance and irrelevance feedback. The user evaluates the retrieved documents and classifies them into five categories:

1. Very Relevant
2. Relevant
3. In-Between
4. Non-relevant
5. Very Non-relevant

**Learning Agent**: The main function of the learning agent is to learn the user's information requirement from the example documents provided by the user, the documents retrieved by EIRS and the relevance feedback given by the user for the documents retrieved. The physical recording of the learning contains a positive user profile and a negative user profile.

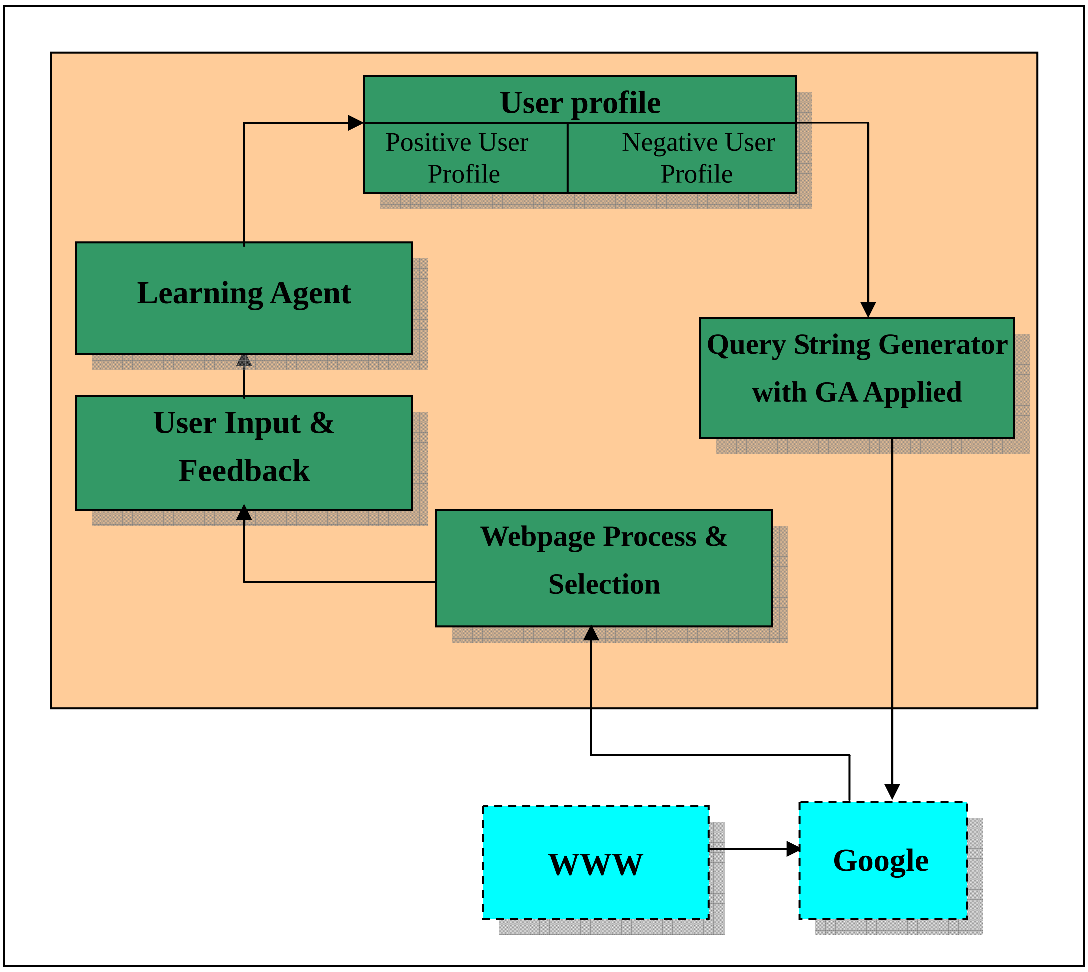


Figure 1. EIRS System framework.

The positive user profile is a database table with three attributes – term, frequency and sensitivity. After the user inputs one example document, the positive user profile with frequency as the term weight is constructed.

Then, EIRS generates a query string and retrieves 10 webpages for the user to evaluate. The user evaluates one webpage and provides relevant feedback to EIRS. Term frequency is determined by Formula (1) and Table 1. After the user evaluates 10 webpages, sensitivity value is generated according to *tas* in relevant/non-relevant documents as shown in Table 2. Frequency is adjusted by formula (3). Then, the positive user profile with Frequency and Sensitivity together as the term weight is constructed, and sorted by Frequency in descending order.

Next, EIRS generates the optimal query string and provides 10 documents for the user to

Table 1. Term frequency adjustment value.

| User's feedback on a retrieved document | C value in formula (1) | Positive/Negative user profile to be inferenced |
|---|---|---|
| Very Relevant | 1.2 | Positive user profile |
| Relevant | 1 | Positive user profile |
| In-Between | 0 | N/A |
| Non-relevant | 1 | Negative user profile |
| Very Non-relevant | 1.2 | Negative user profile |

Table 2. Strategy of adjusting sensitivity from feedback documents

| Conditions | Sensitivity value or action |
|---|---|
| The term appeared in the relevant document and not appeared in the non-relevant document | 1.2 |
| The term appeared in the relevant document and appeared in the non-relevant document | 1<br>Remove the negative term from negative user profile |

evaluate. After the user has rated one retrieved document as relevant, Frequency is adjusted again by formula (1) and Table 1.

After the user evaluates 10 webpages, sensitivity value is generated by Table 2. Frequency is adjusted by formula (3). Finally, positive user profile is modified and sorted again by Frequency in descending order. The negative user profile is a database table with two attributes – term and frequency.

After the user inputs one example document, EIRS generates query string and retrieves 10 webpages for the user to evaluate. The user evaluates one webpage and provides non-relevant feedback to EIRS. NFrequency, adjusted non-interested term frequency, is determined by Formula (2) and Table 1. Before the user profile is generated, NFrequency in negative user profile has no initial value. Hence, NFrequency is determined by $F_R$, term frequency in a retrieved document, and adjustment value C. After the user evaluates 10 webpages, negative user profile is generated and sorted by NFrequency in descending order. Next, EIRS generates the optimal query string and provides 10 documents for the user to evaluate. After the user has rated one retrieved document as non-relevant, NFrequency is adjusted again by formula (2) and Table 1. After the user evaluates 10 webpages, negative user profile is modified. Finally, negative user profile is sorted again by NFrequency in descending order.

Like most searchers setting the constant to a value for their experiments after pretest [5, 11, 18, 33], C value in Table 1 and sensitivity value in Table 2 are decided after our preliminary tests.

(1) Frequency = Frequency ＋ C × $F_R$

Where

Frequency: initial frequency/adjusted term frequency in positive user profile

C: adjustment value based on the user's feedback for a retrieved document as Table 1.

$F_R$: term frequency in a retrieved document

(2) NFrequency = NFrequency ＋ C × $F_R$

Where

NFrequency: adjusted term frequency in negative user profile

C: adjustment value based on the user's feedback for a retrieved document as Table 1.

$F_R$: term frequency in a retrieved document

(3) Frequency = Frequency × Sensitivity

Where

Frequency: adjusted term frequency in positive user profile

Sensitivity: the sensitivity value based on the strategy of adjusting sensitivity as Table 2.

**User Profile**: There are two user profiles maintained, the positive user profile and the negative user profile, to record the system's learning of the user's information requirements. The positive user profile is used to record the system's learning of the user's interest. Table 3 shows the data structure for the positive user profile. The negative user profile is used to record the system's learning of the user's disinterest. Table 4 shows the data structure for the negative user profile. Terms in both user profiles will be sorted by the Frequency/NFrequency value in descending order. The top 100 terms in the positive user profile is used to manifest the system's final learning of the user's interest. The top 100 terms in the negative user profile is used to manifest the system's final learning of the user's disinterest.

**Query String Generator with GA Applied**: The main function of this module is to generate a query string from the key words selected by the GA Agent according to the chromosome generated. In this module, each chromosome consists of 20 bits, 16 bits for positive keywords from positive user profile connected by AND operator and 4 bits for negative keywords from negative user profile connected by NOT operator. Each bit represents one keyword selected or not selected. When the value of a bit equals to 1, it represents this keyword is selected in this chromosome. When the value of a bit equals to 0, it represents this keyword is not selected. At last, GA produces optimal chromosome for query string.

**Webpage Processing and Selecting**: The main function of this module is to compare the similarity between the document retrieved and the positive user profile first, then pass the similarity value to the GA in another module, and finally generate 10 most relevant documents to the user. The EIRS we have developed is based on vector space model. According to vector space model, the user profile and the document are thought of as vectors in an n-dimensional space, where each dimension represents an index term with a weight.

Tf-idf developed by Gerard Salton [28] is the most common term weighting scheme. However, in this study, we have adopted the Nick's term weighting scheme to avoid the

Table 3. Data structure of the positive user profile

| Field name | Type | Descriptions |
|---|---|---|
| Term | String | Terms from relevant documents |
| Frequency | Number | initial frequency/adjusted term frequency |
| Sensitivity | Number | Situations of term appearance in relevant/non-relevant documents |

Table 4. Data structure of negative user profile

| Field name | Type | Descriptions |
|---|---|---|
| NTerm | String | Terms from non-relevant documents |
| NFrequency | Number | Adjusted non-interested term frequency |

requirement of providing a sufficient number of examples before the learning agent can begin the process of searching for new information in Salton's weighting scheme. Nick's scheme was modified from Salton's tf-idf. In Nick's study, all keywords that held the greater weight from all the text documents that the user provided as examples were merged in a file called Dictionary. The representation of the Dictionary was an $N \times 3$ matrix, where $N$ is the number of keywords. The first column of the matrix contained the keywords, the second column was the total number of documents that contained the keywords, and the last column was the sum of the keyword's frequencies in all the texts that appeared. The Dictionary's keywords were sorted according to their weight, which was given by the following formula [23]:

$$w_i = \frac{\left(\frac{freq_i}{freq_{\max}}\right)}{\sqrt{\sum_{j=1}^{N}\left(\frac{freq_j}{freq_{\max}}\right)^2}}$$

Where

$freq_i$: the frequency of the keyword $i$ in all texts in which it appear;

$freq_{\max}$: the maximum keyword frequency of all keywords in the Dictionary;

$N$: the number of keywords in the Dictionary.

In vector space model, cosine angle between two documents represented as vectors is the most popular approach to compare the similarity between two documents. The formula is as follows:

$$sim(D,Q) = \cos(\theta) = \frac{\vec{D}\cdot\vec{Q}}{|\vec{D}||\vec{Q}|}$$

$$= \frac{D \bullet Q}{|D| \times |Q|} = \frac{\sum_{i=1}^{n} w_{Di} \times w_{Qi}}{\sqrt{\sum_{i=1}^{n} w_{Di}^2} \times \sqrt{\sum_{i=1}^{n} w_{Qi}^2}}$$

Where

Sim ($Q$, $D_i$) = similarity between Document $D_i$ and Query $Q$

$D$ — Document

$Q$ — Query

$w_{Di}$ — weight of term i in document $D$

$w_{Qi}$ — weight of term i in query $Q$

If the document and the user profile are very similar, the angle should be very small. On the other hand, if the angle is very high, the vectors would be close to perpendicular and the cosine angle would be 0. In short, cosine (90) = 0 (completely unrelated); cosine (0) = 1 (completely related).

The vector space model has the advantage of producing a ranked list of documents based on their similarities to the query. After computing the similarities between the user profile and the retrieved documents, this module will produce a ranked 10 documents to the user.

## 3.2 System flow of EIRS

The system flows of EIRS are as follows:

Step 1: Input one example document in the User Input and Feedback module. The positive user profile with frequency as the term weight constructed.

Step 2: Generate query string, return 10 webpages for users to evaluate.

Step 3: Evaluate the 10 webpages, and provide relevant/non-relevant feedback to the EIRS system in the User Input and Feedback module.

Step 4: Learn user's interest and disinterest from user's relevant and non-relevant feedback, and weight the terms.

Step 5: Create the positive user profile and negative user profile with frequency and sensitivity together as term weight revised. Positive user profile is sorted by adjusted term frequency in descending order and negative user profile is sorted by adjusted non-interested term frequency in descending order.

Step 6: Select top 16 positive terms from positive user profile by Frequency order. Select top 4 negative terms from negative user profile by NFrequency order. These 20 selected terms passed to GA to generate the optimal query string, select 10 documents most close to the user profile, and provide these 10 documents to the user for relevance feedback.

Step 7: Evaluate the 10 documents as (very) relevant or (very) non-relevant and feedback to the EIRS.

Step 8: Learn user's interest and disinterest from user's relevant and non-relevant feedback, and re-weight the terms.

Step 9: Modify positive user profile and negative user profile. Positive user profile is sorted by adjusted term frequency in descending order and negative user profile is sorted by adjusted non-interested term frequency in descending order.

Step 10: Select top 16 positive terms from positive user profile by Frequency order. Select top 4 negative terms from negative user profile by NFrequency order. These 20 selected terms passed to GA to generate the optimal query string, select 10 documents most close to the user profile, and provide these 10 documents to the user for relevance feedback.

Step 11: Evaluate the 10 documents as (very) relevant or (very) non-relevant and feedback to the EIRS.

The algorithm of modifying positive user profile and negative user profile according to the user's relevance feedback is as follows:

**Input:** positive/negative user profile, retrieved documents

**Output:** Modified positive user profile with sensitivity set for each term, modified negative user profile

1. For I = 1 to 10
2. Case (the user's relevance rating on the retrieved document)
3. Case 1: Very Relevant
4. Frequency = Frequency $+$ $1.2 \times F_R$
5. Case 2: Relevant
6. Frequency = Frequency $+$ $F_R$
7. Case 3: In-Between
8. No action
9. Case 4: Non-relevant
10. NFrequency = NFrequency $+$ $F_R$
11. Case 5: Very Non-relevant
12 NFrequency = NFrequency $+$ $1.2 \times F_R$
13 End Case
14. Next I
15. If (the Term appeared in relevant documents but not appeared in the non-relevant documents)
16. Then
17. Set Sensitivity = 1.2
18. Frequency = Frequency × Sensitivity

19. Else (the Term appeared in relevant documents and also appeared in the non-relevant documents)
20.     Set Sensitivity = 1
21.     Remove the Term from the negative user profile
22. Endif

# 4. Experiments and Results

To study the effect of the extraction and application of the information of *tas*, we have designed and conducted 2 experiments. Before that, 3 pre-experiments were done first to detect the appropriate system variables. We have selected twenty persons possessing a minimum of a bachelor's degree and five years of web search experience as the testee.

In our preliminary assessments, we find that the amount of example documents has a positive effect on EIRS performance. The more example documents the user provides, the better performance EIRS has. Since the primary purpose of this research is to explore the effect of the extraction and application of the information of *tas*, the beginning performance of the system is not the major concern. Therefore, providing one example document at the beginning is selected because it is easier and convenient to the user. In another preliminary assessment, we discover the performance of the amount of document retrieved equaling to 5 is better than 10.

Nevertheless, the beginning low performance could be improved after user's feedback. We find the optimal amounts of relevant feedback documents is about 6 to 7 and non-relevant feedback documents is about 3 to 4 for conducting the experiments of the effect of sensitivity and the effect of negative user profile when the amount of document retrieved equals to 10. According to the result from the preliminary assessments, the system configuration of the EIRS is set to one example document provided by users and 10 ranked retrieved documents.

## 4.1 Experimental process

The experimental processes and the formation of the user profiles are as follows:

1) The user inputs one example document to EIRS. (The positive user profile with frequency as the term weight constructed.)

2) EIRS processes the input from step 1 and output the URLs of the retrieved documents.

3) The user browses the retrieved documents and ranks each as “Very Relevant“, “Relevant”, “In-Between”, “Non-relevant” or “Very Non-relevant” and inputs the retrieved documents with ranks to EIRS. (The positive user profile with frequency & sensitivity together as the term weight constructed; the negative user profile with frequency & sensitivity together as the term weight constructed.)
4) EIRS processes the input from step 3 and output the URLs of the retrieved documents.
5) The user browses the retrieved documents and ranks each as “Very Relevant“, “Relevant”, “In-Between”, “Non-relevant” or “Very Non-relevant” and inputs the retrieved documents with ranks to EIRS. (The positive user profile with frequency & sensitivity together as the term weight revised; the negative user profile with frequency & sensitivity together as the term weight revised.)
6) EIRS processes the input from step 5 and output the URLs of the retrieved documents.
7) The user browses the retrieved documents and ranks each as “Very Relevant“, “Relevant”, “In-Between”, “Non-relevant” or “Very Non-relevant” as the basis to calculate rate of correctness.

The experimental processes are designed to have the positive and negative user profiles constructed and revised once. Working with these experimental processes, we have conducted 3 pre-experiments to detect the appropriate system variables and 2 experiments to explore the effect of sensitivity and negative user profile. Table 5 shows the 5 experiments and the variables to be manipulated. Every experiment explores one variable and has the other variables controlled.

### 4.2 Experimental Results

There are five experiments in this study. The processes and the results of these 5 experiments are described as follows.

Experiment 1, 2 and 3 are about detecting the appropriate system variables. The processes

Table 5 The experiments and the variables.

| Experiment | Variable | Values of the variable |
|---|---|---|
| 1 | Amount of terms of the user profile | 30/ 50/ 100/ 150/ 200 |
| 2 | Amount of key words represented by a chromosome | 10/ 15/ 20/ 25/ 30 |
| 3 | Amount of negative terms used as key words. | 2/ 4/ 6 |
| 4 | Strategy of term weighting | Frequency/ frequency together with sensitivity |
| 5 | Strategy of user profile | 20,0/ 16,4 |

and the results of these experiments will be discussed in section 4.2.1.

Experiment 4 is about studying the effect of sensitivity. We want to compare the effect difference between term weighting strategy based on frequency and our strategy based on frequency together with sensitivity. The process and the result of experiment 4 will be discussed in section 4.2.2.

Experiment 5 is about studying the effect of negative user profile. We want to compare the effect difference between positive user profile with negative user profile and positive user profile without negative user profile. The process and the result of experiment 5 will be discussed in section 4.2.3.

### 4.2.1 Experiments for system factor adjustment

**Experiment 1**: The purpose of this experiment is to explore the effect of amount of terms of user profile. The experimental variable is the amount of terms of user profile. Other variables are controlled, the variable of amount of key words represented by a chromosome is set to 20; the variable of strategy of user profile is set to positive user profile with negative user profile, the amount of positive and negative terms used as key word is 16, 4; and the variable of term weighting strategy is set to frequency together with sensitivity. Figure 2 is the experimental result. Axis X marks the value of the experimental variable; axis Y marks EIRS performance. The result shows that EIRS performance increases first as the amount of terms in user profile increases, then it begins to drop as the terms in user profile reaches a certain amount.

The result shows that EIRS performance has the best performance, 64%, when the amount of terms of user profile equals to 100. According to the result from this experiment, the quantity of 100 terms of the user profile will be set to the best system configuration.

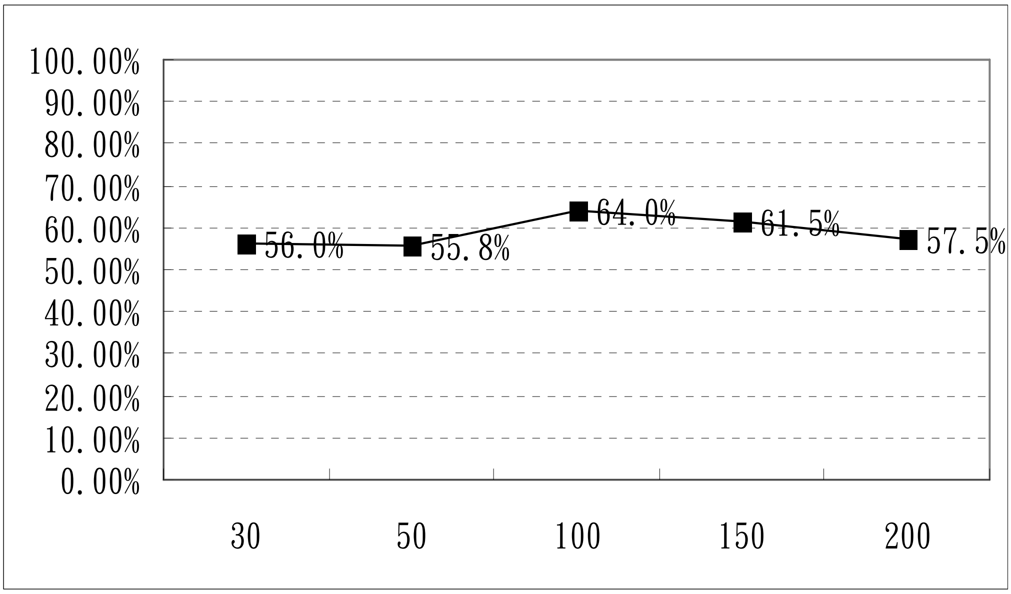


Fig.2 The effect of amount of terms of user profile

**Experiment 2**: The purpose of this experiment is to explore the effect of the amount of key words represented by a chromosome. The experimental variable is the amount of key words represented by a chromosome. The key words are all selected from positive user profile and not any key words from negative user profile. Other variables are controlled, the variable of the amount of terms of user profile is set to 100, and the variable of term weighting strategy is set to frequency together with sensitivity. Figure 3 is the result of experiment 2. Axis X marks the value of the experimental variables; axis Y marks EIRS performance. The result shows that there is no obvious performance change when the amounts of key words represented by a chromosome are different. Consider the performance and system effectiveness, the quantity of 20key words are selected to be represented by a chromosome.

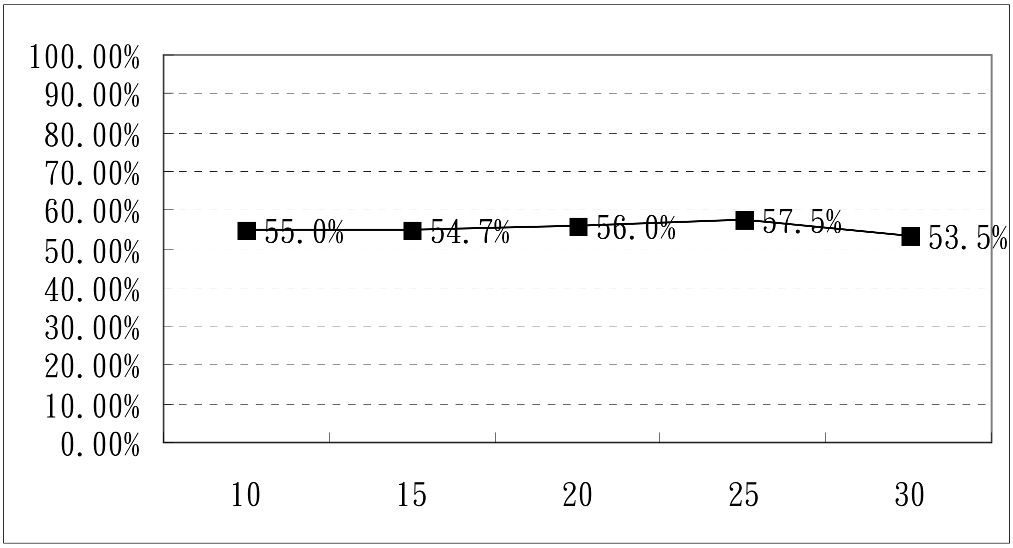


Fig.3 The effect of the amount of key words represented by a chromosome

**Experiment 3**: The purpose of this experiment is to explore the effect of keyword organization. The experimental variable is amount of positive and negative terms used as key words. Other variables are controlled, the variable of the amount of terms of user profile is set to 100, the variable of amount of key words represented by a chromosome is set to 20, and the variable of term weighting strategy is set to frequency together with sensitivity. Values for the experimental variable are 18,2, 16,4, and 14,6. Fig.4 is the experimental result. Axis X marks the value of the experimental variables. Axis Y marks the EIRS performance. The result shows that there is no significant difference in performance among the different amount of negative term used as key words. However, the result reveals that negative terms used in the NOT operator could have better performance than positive only, 56%. At here, this research just selects the best keyword organization (keyword organization formed by 16 positive and 4 negative terms) according to the result for the best system configuration.

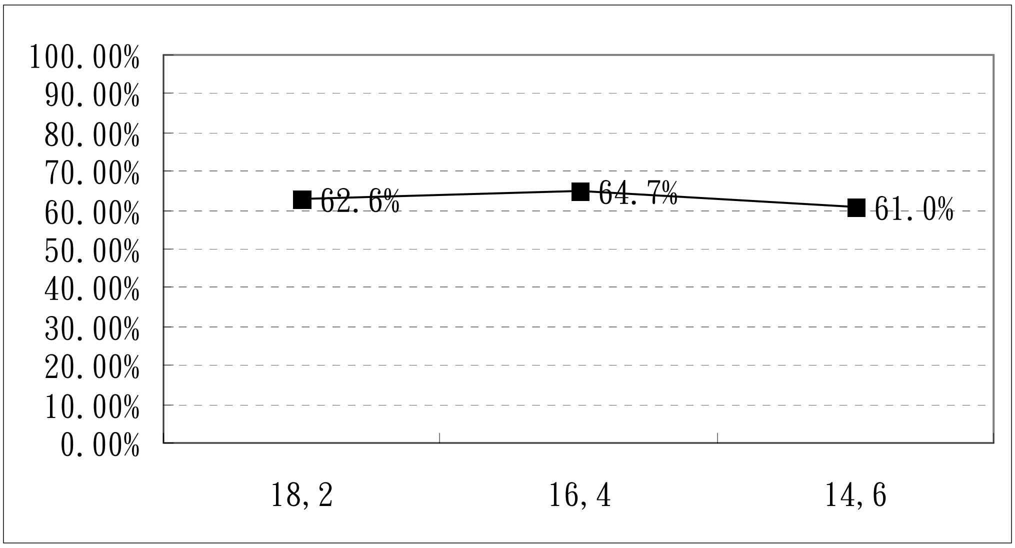


Fig.4 The effect of keyword organization

### 4.2.2 Study of the effect of sensitivity

**Experiment 4**: The purpose of this experiment is to study the effect of sensitivity. The terms belonging to *tas* 1 and 3 will be input into positive user profile. However, the terms belonging to *tas* 1 should be more important than ones belonging to *tas* 3 to the user. Sensitivity is for adjusting and distinguishing the term weight. This experiment is to apply the sensitivity and study the effect of sensitivity. In this experiment, the experimental variable is term weighting strategy. other variables are controlled, the variable of the amount of terms of user profile is set to 100, the variable of amount of key words represented by a chromosome is set to 20, and the variable of strategy of user profile is set to positive user profile with negative user profile, the amount of positive and negative terms used as key word is 16,4. Figure 5 is the experimental result. Axis X marks the value of the experimental variables. Axis Y marks EIRS performance. Values for the experimental variable are frequency and frequency together with sensitivity. Frequency strategy weights the term according to the sorted sequence of frequency. Formula (1) is used in the strategy of term weighting of frequency. The strategy of frequency weights the term according to the sorted sequence of the frequency. Formula (3) is used in the strategy of term weighting of frequency together with sensitivity. The strategy of frequency together with sensitivity weights the term according to the sorted sequence of the product of frequency and sensitivity. Figure 5 shows that term weighting strategy based on frequency together with sensitivity has a better effect than strategy based on frequency only. This experiment also demonstrates how to use the sensitivity information, and this experimental result reveals the sensitivity information used in EIRS is useful and effective.

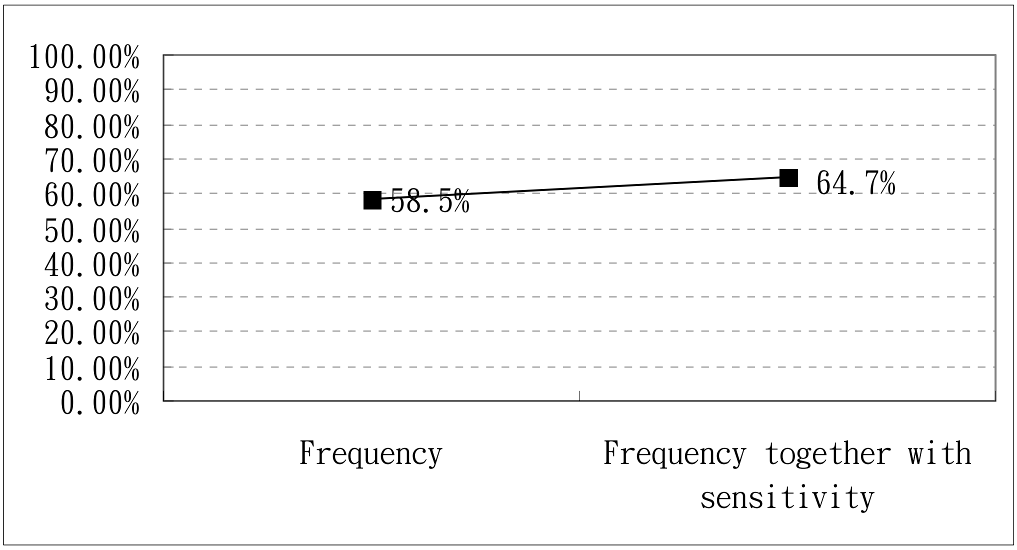


Fig.5 The effect of strategy of term weighting

### 4.2.3 Study of the effect of negative user profile

**Experiment 5**: The purpose of this experiment is to study the effect of negative user profile. In EIRS, the user feedbacks are classified into Very Relevant, Relevant, In-Between, Non-relevant, and Very Non-relevant. The feedbacks of Very Relevant, Relevant are about positive feedback. EIRS finds positive terms from positive feedback and establishes the positive user profile. The feedbacks of Non-relevant and Very Non-relevant are about negative feedback. EIRS finds negative terms from negative feedback and establishes the negative user profile. The terms belonging to *tas* 2 are user's disinterests as aforementioned. Negative user profile is for keeping these user's disinterests. This experiment is to apply the information of negative user profile and study its effect. To study the effect of negative user profile, we compare the performance between positive user profile with negative user profile and positive user profile without negative user profile. The experimental variable is the strategy of user profile. Values for the experimental variable of the strategy of user profile are amount of positive and negative terms used as key words, 20,0 and 16,4. Positive terms are from the positive user profile and connected by AND operator. Negative terms are from the negative user profile and connected by NOT operator. In this experiment, other variables are controlled, the variable of the amount of terms of user profile is set to 100, the variable of amount of key words represented by a chromosome is set to 20, and the variable of term weighting strategy is set to frequency together with sensitivity. Figure 6 is the experimental result. Axis X marks the value of the experimental variables. Axis Y marks the EIRS performance. Figure 6 shows that the quantity 16,4 of positive and negative term used as key words has a better effect than having 20 positive terms only. According to the result from this experiment, the strategy of positive user profile with negative user profile has better performance than the strategy of positive user profile only on EIRS. This experiment also demonstrates how to use the information of negative user profile, and the experimental result of this experiment reveals the information of negative user profile used in EIRS is useful and effective.

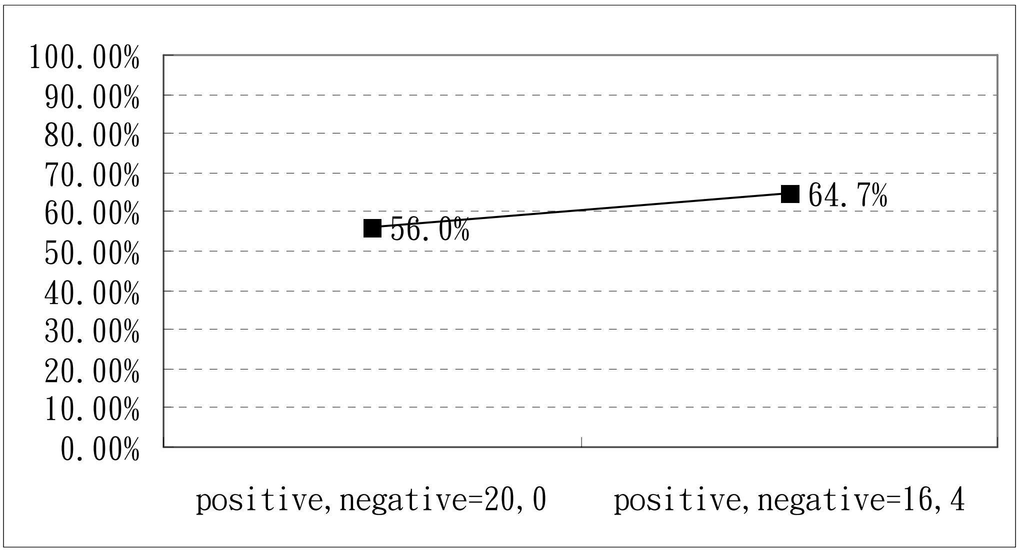


Fig.6 The effect of strategy of user profile.

# 5. Conclusion

This study has identified the information of *tas* in the rated relevant/non-relevant documents. A method together with an IR system is developed to demonstrate the extraction and application of the information. Experiments have also been conducted to study its effect. The experimental results preliminarily show that the information of *tas* could be extracted and appropriately applied to enhance retrieval effectiveness. First, the information of *tas* could be used together with term frequency to form and weight the vector expression of the user's information interest to provide 'AND' query string generation basis and to be used in the similarity comparison with the retrieved document. Second, the information of *tas* could be used to form the vector expression of the user's disinterest to provide 'NOT' query string generation basis. However, the study of *tas* in this research is an initial exploration. Optimal configurations need to be sought and the values of some parameters need to be finely-tuned in the future study.

Our study of *tas* is not to compare with Rocchio's approach and the related studies. It is complementary to the existing product instead of replacing. For instance, the information of *tas* 2 could be combined with Rocchio's formula to decrease the number of non-relevant documents retrieved. Furthermore, the information identified and extracted in this study also can be used in various feedback applications of IR systems.

Future work needs to be done to determine the appropriate value setting for the sensitivity. Factors to be considered could include the term appearance frequency and distribution under *tas*. In addition, as the application of *tas* has been shown to have impact on the retrieval effectiveness, additional exploring could be considered. One possible application is to have the terms belonging to *tas* 2 and 3 used in the vector to express the user's disinterest. These terms could be used with different importance as the basis in the similarity comparison with other vectors. It could be used to filter out the disinterested document retrieved by the IR system as interested document.

# References


[1] Azimi-Sadjadi, M., Salazar, J., Srinivasan, S., and Sheedvash, S., "An adaptable connectionist text retrieval system with relevance feedback", Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Vol. 1, pp. 309-314, Budapest, Hungary, July 2004.

[2] Balabanovic, M., "An Adaptive Web Page Recommendation Service", Proceedings of the First International Conference on Autonomous Agents, pp. 378-385, New York, February 1997.

[3] Biron, P., and Kraft, D., "New Methods for Relevance Feedback: Improving Information Retrieval Performance", Proceedings of the ACM Symposium on Applied Computing, pp. 482-487, Nashville, Tennessee, February 1995.

[4] Buckley, C. and Salton, G., "Optimization of relevance feedback weights", Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 351-357, Seattle, Washington, 1995.

[5] Choi, J., Kim, M., and Raghavan, V., "Adaptive relevance feedback method of extended Boolean model using hierarchical clustering techniques", Information Processing and management, Vol. 42, No. 2, pp. 331-349, March 2006.

[6] Christiansen, A., and Lee, D., "Relevance feedback query refinement for PDF medical journal articles", Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, pp. 57-62, Salt Lake City, Utah, June 2006.

[7] Crestani, F., "Neural relevance feedback for information retrieval", In B. Bouchon-Meunier, B., Yager, R., & Zadeh, L., (Eds.), Uncertainty in intelligent in information systems, World Scientific, Singapore, 2000.

[8] Desjardins, G. and Godin, R., "Combining Relevance Feedback and Genetic Algorithms in an Internet Information Filtering Engine", Proceedings of RIAO2000 Conference, Vol. 2, pp. 1676-1685, Paris, France, April 2000.

[9] Dillon, M. and Desper, J., "Automatic Relevance Feedback in Boolean Retrieval System", Journal of Documentation, Vol. 36, pp. 197-208, 1980.

[10] Drucker, H., Shahary, B., and Gibbon, D., "Relevance Feedback using Support Vector Machines", Proceedings of the 18th International Conference on Machine Learning (ICML) , pp. 122-129, Williamstown, MA, June 2001.

[11] Ekkelenkamp, R., Kraaij, W., and Leeuwen, D., "TNO TREC7 Site Report: SDR and Filtering", Proceedings of the Seventh Text REtrieval Conference, pp. 455-462, Gaithersburg, Maryland, November 1998.

[12] Graugaard, L., "Implicit relevance feedback in interactive music: issues, challenges, and case studies", Proceedings of the 1st international conference on Information interaction in context, pp. 119-128, Copenhagen, Denmark, October 2006.

[13] Harman, D., "Relevance Feedback Revised", Proceedings of 15th Annual International

ACM SIGIR Conference, pp. 1-10, New York, June 1992.

[14] Hoashi, K., Matsumoto, K., Inoue, N., and Hashimoto, K., "Document Filtering Method Using Non-Relevant Information Profile", Proceedings of ACM SIGIR 2000, pp. 176-183, Athens, Greece, July 2000.

[15] Hoashi, K., Zeitler, E., Inoue, N., "Implementation of Relevance Feedback for Content-based Music Retrieval Based on User Preferences", Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 385-386, Tampere, Finland, August 2002.

[16] Hoeber, O., and Yang, X., "Interactive Web Information Retrieval Using WordBars", Proceedings of 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 875-882, Hong-Kong, December 2006.

[17] Ide, E., "New Experiments in Relevance Feedback", In Salton G.(Ed.), The SMART Retrieval System: Experiments in automatic Document Processing, pp. 337-354, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[18] Justino, E., Bortolozzi, F., and Sabourin, R., "A comparison of SVM and HMM classifiers in the off-line signature verification", Pattern Recognition Letters, Vol. 26, No. 9, pp. 1377-1385, July 2005.

[19] Kim, B., Kim, J., and Kim, J., "Query term expansion and reweighting using term co-occurrence similarity and fuzzy inference", Proceedings of IFSA World Congress and 20th NAFIPS International Conference, Vol. 2, pp. 715-720, Vancouver, Canada, July 2001.

[20] Li, F., Mehlitz, M., Feng, L., and Sheng, H., "Web Pages Clustering and Concept Mining: An Approach Intelligent Information Retrieval", Technical Program of 2006 IEEE International Conferences on Cybernetics & Intelligent Systems (CIS) and Robotics, Automation & Mechatronics (RAM), Bangkok, Thailand, June 2006.

[21] Moyotl, E., and Jimenez, H., "An Analysis on Frequency of Terms for Text Categorization", Proceedings of Conference of Spanish Natural Language Processing Society, pp. 141-146, Barcelona, July 2004.

[22] Navigli, R., and Velardi, P., "An analysis of ontology-based query expansion strategies", Proceedings of Workshop on Adaptive Text Extraction and Mining in the 14th ECML, pp. 42-49, Croatia, September 2003.

[23] Nick, Z., and Themis, P., "Web Search Using a Genetic Algorithm", IEEE Internet Computing, Vol. 5, No. 2, pp. 18-26, March/April 2001.

[24] Ng, H., Ang, H., and Soon, W., "DSO at TREC-8: A Hybrid Algorithm for the Routing Task", Proceedings of the Eighth Text REtrieval Conference (TREC-8), pp. 267, Gaithersburg, Maryland, November 1999.

[25] Rho, S., Hwang, E., and Kim, M., "Music Information Retrieval Using a GA-based Relevance Feedback", Proceedings of 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07) , pp. 739-744, Seoul, Korea, April, 2007.

[26] Robertson, S., and Sparck-Jones, K., "Relevance Weighting of search terms", Journal of the American Society for Information Science, Vol. 27, No.3, pp. 129-146, May/June 1976.

[27] Rocchio, J., Document retrieval systems – Optimization and evaluation, Unpublished doctoral dissertation, Harvard University, Cambridge, MA, USA, March 1966.

[28] Salton, G., and Buckley, C., "Term weighting approaches in automatic text retrieval", Information Processing and Management, Vol. 24, pp. 513-523, Nov. 1988.

[29] Salton, G., Fox, E., and Wu, H. "Extended Boolean information retrieval", Communication of the ACM, Vol. 26, No.11, pp. 1022-1036, November 1983.

[30] Savoy, J., "Data Fusion for Effective European Monolingual Information Retrieval", Working Notes for the Cross Language Evaluation Forum (CLEF) 2004 Workshop, pp. 233-244, Bath, UK, September 2004.

[31] Shin,K., Han, S., Gelbukh, A., and Park, J., "Advanced Relevance Feedback Query Expansion Strategy for Information Retrieval in MEDLINE", LNCS Vol. 3287, pp. 425-431, October 2004.

[32] Singhal, A., Mitra, M., and Buckley, C., "Learning routing queries in a query zone", Proceedings of the Twentieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 25-32, Philadelphia, July 1997.

[33] Vires, A., and Roelleke, T., "Relevance Information: A Loss of Entropy but a Gain for IDF?", Proceedings of SIGIR'05, pp. 282-289, Salvador, Brazil, August 2005.

[34] Wei, C., and Li, C., "Learning Pathological Characteristics from User's Relevance Feedback for Content-Based Mammogram Retrieval", Proceedings of Eighth IEEE International Symposium on Multimedia (ISM'06), pp. 738-741, San Diego, CA, December 2006.

[35] Xu, X., Lee, D., Antani, S., and Long, L., "Relevance feedback for spine X-ray retrieval", Proceedings of the 18th International Symposium on Computer-Based Medical Systems, pp. 197-202, Dublin, Ireland, June 2005.

## APPEDIX A: Stopword List

| | | | |
|---|---|---|---|
| A | ABOUT | ABOVE | ACROSS |
| AFTER | AFTERWARDS | AGAIN | AGAINST |
| ALL | ALMOST | ALONE | ALONG |
| ALREADY | ALSO | ALTHOUGH | ALWAYS |
| AMONG | AMONGST | AN | AND |
| ANOTHER | ANY | ANYHOW | ANYONE |
| ANYTHING | ANYWHERE | ARE | AROUND |
| AS | AT | BE | BECAME |
| BECAUSE | BECOME | BECOMES | BECOMING |
| BEEN | BEFORE | BEFOREHAND | BEHIND |
| BEING | BELOW | BESIDE | BESIDES |
| BETWEEN | BEYOND | BOTH | BUT |
| BY | CAN | CANNOT | CO |
| COULD | DOWN | DURING | EACH |
| EG | EITHER | ELSE | ELSEWHERE |
| ENOUGH | ETC | EVEN | EVER |
| EVERY | EVERYONE | EVERYTHING | EVERYWHERE |
| EXCEPT | FEW | FIRST | FOR |
| FORMER | FORMERLY | FROM | FURTHER |
| HAD | HAS | HAVE | HE |
| HENCE | HER | HERE | HEREAFTER |
| HEREBY | HEREIN | HEREUPON | HERS |
| HERSELF | HIM | HIMSELF | HIS |
| HOW | HOWEVER | I | IE |
| IF | IN | INC | INDEED |
| INTO | IS | IT | ITS |
| ITSELF | LAST | LATTER | LATTERLY |
| LEAST | LESS | LTD | MANY |
| MAY | ME | MEANWHILE | MIGHT |
| MORE | MOREOVER | MOST | MOSTLY |
| MUCH | MUST | MY | MYSELF |
| NAMELY | NEITHER | NEVER | NEVERTHELESS |
| NEXT | NO | NOBODY | NONE |
| NOONE | NOR | NOT | NOTHING |
| NOW | NOWHERE | OF | OFF |
| OFTEN | ON | ONCE | ONE |
| ONLY | ONTO | OR | OTHER |

| | | | |
|---|---|---|---|
| OTHERS | OTHERWISE | OUR | OURS |
| OURSELVES | OUT | OVER | OWN |
| PER | PERHAPS | RATHER | SAME |
| SEEM | SEEMED | SEEMING | SEEMS |
| SEVERAL | SHE | SHOULD | SINCE |
| SO | SOME | SOMEHOW | SOMEONE |
| SOMETHING | SOMETIME | SOMETIMES | SOMEWHERE |
| STILL | SUCH | THAN | THAT |
| THE | THEIR | THEM | THEMSELVES |
| THEN | THENCE | THERE | THEREAFTER |
| THEREBY | THEREFORE | THEREIN | THEREUPON |
| THESE | THEY | THIS | THOSE |
| THOUGH | THROUGH | THROUGHOUT | THRU |
| THUS | TO | TOGETHER | TOO |
| TOWARD | TOWARDS | UNDER | UNTIL |
| UP | UPON | US | VERY |
| VIA | WAS | WE | WELL |
| WERE | WHAT | WHATEVER | WHEN |
| WHENCE | WHENEVER | WHERE | WHEREAFTER |
| WHEREAS | WHEREBY | WHEREIN | WHEREUPON |
| WHEREVER | WHETHER | WHITHER | WHICH |
| WHILE | WHO | WHOEVER | WHOLE |
| WHOM | WHOSE | WHY | WILL |
| WITH | WITHIN | WITHOUT | WOULD |
| YET | YOU | YOUR | YOURS |
| YOURSELF | YOURSELVE | | |