



ELSEVIER

Available online at www.sciencedirect.com

Decision Support Systems xx (2004) xxx–xxx

Decision Support
Systemswww.elsevier.com/locate/dsw

Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison

Weiguo Fan^{a,*}, Michael D. Gordon^b, Praveen Pathak^c

^a *Virginia Tech, Accounting and Information Systems, 3007 Pamplin Hall 24061, Blacksburg, VA, USA*

^b *University of Michigan, USA*

^c *University of Florida, USA*

Received 17 April 2003; received in revised form 23 January 2004; accepted 3 February 2004

Abstract

Due to the overwhelming volume of information that is increasingly available, many people rely on current awareness systems to keep abreast of the latest developments in the fields that they are interested in, as evidenced in the popularity of subscriptions to news-monitoring and digital library services. The success of these services, however, often requires effective acquisition of users' personal standing interests as represented in personal profiles. Our objective in this paper is twofold. First, we have introduced a new method for profile generation and compared it against other well-known methods. We have found promising results. Second, although there are various methods proposed in information retrieval and machine learning literature to address the issue of profiling, a unified framework and systematic cross-system comparison to help users, especially service providers, to determine the most effective way of profiling consumers is still lacking in the literature. In this paper, we try to fill the gap by looking at these methods from a more integrated point of view based on statistical contingency theory. Variations of these methods are then systematically tested on three well-known routing systems and results are analyzed and reported.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Personalization; Profiling; Information push; Information routing; Selective dissemination of information; Information retrieval

1. Introduction

Successful decision making calls for timely information. Business managers need to continue gathering information about consumers, competitors, and markets to formulate and modify their business plan and business strategies. Scientists require current

awareness systems from digital libraries to keep abreast of the latest developments in their corresponding fields and to avoid replicating the work done by other peers. Even common people who invest in stocks rely on all kinds of research reports from financial investing institutions to monitor companies' financial performance and breaking news.

All in all, there is huge demand of information from all sectors of business. In order to fulfill the needs of information from such a large group of customers, web portals, information service providers, digital libraries, news-wire companies, etc., are build-

* Corresponding author. Tel.: +1-540-231-6588; fax: +1-540-231-2511.

E-mail address: wfan@vt.edu (W. Fan).

ing large-scale systems that can help them distribute or deliver timely information to their customers' email boxes, and mobile devices like digital cellular phone, and PDAs. This kind of service is commonly called *Selective Dissemination of Information* (SDI) [16] or *Information Routing*. In the current e-Business era, it is often referred to as *Push Technology* [7].¹ The technique that enables companies to deliver these highly customized and personalized information to consumers is the so-called *personalization* technology. The core of the personalization technology is the management of user profiles. These profiles can be implicit profiles that are built from consumers' purchase history, such as the categories of products that consumers purchased recently. Or they could be profiles specified by users explicitly when they initially register or subscribe to the services.

We look at the problem of pushing personalized information goods (news, abstracts, etc.) in this paper. Unlike the physical goods, information goods are intangible and the cost of ownership is often very low (even free). The lack of structure and regularity of online information makes the problem of information push really hard. Without careful design and implementation, the large amount of irrelevant information pushed to these users will often frustrate the users, possibly turning them away from using the services.

Similar to user profiles for tangible goods, there are two different ways of creating user profiles for information goods: an explicit profile which is provided by a user directly, or an implicit profile which is built by a push system based on a user's feedback and behavior tracking (with appropriate user consent). Instead of having past purchase history (constituting explicit profile), past reading history, bookmarks, and news folders are available for implicit user profile generation.

We concentrate our effort on implicit profile generation methods because explicit profiles in information push suffer the same problem as in traditional online information search using generic search engines (Google, Yahoo, etc.): the vocabulary problem [8], where a user often has problem expressing his/her interest using the right words in

getting the information (s)he wants. Often, the words are too few and not expressive enough to represent his/her interests as in most web search scenarios, where the average number of words in search queries is only 2–3 [10]. Also, users often express the conceptual content of his/her interests with query words that do not match the words in relevant documents.

This paper is concerned with implicit profile building used in online push services or information routing, in which a set of keywords built from a user's feedback or past history is used to represent the interests of consumers. Other ways of representing profiles, like using categories or subject headings, are not of our concern in this paper. Moreover, we will look at the profile learning problem from information retrieval perspective since most of the push or routing techniques originated from the information retrieval research.

To create a user profile, most routing algorithms learn a set of features that may potentially help distinguish relevant documents from non-relevant documents. Based on the occurrences of these features in a new document, the new document either can be considered potentially useful and is routed to the user or can be considered non-relevant and is discarded. Most current systems also assign weights to these features to indicate the importance of these features for relevance estimation.

It is to be noted that in traditional information retrieval research the routing task is split into (1) selecting the terms for the profile and (2) using the profile through routing functions. Some recent approaches [3] using support vector machines do not need such splitting of tasks. But in this paper we follow the task splitting model utilized in traditional information retrieval.

To learn the features and their weights, most routing algorithms typically use the probability of occurrences (or some variations of it) of a feature in the documents marked relevant (and non-relevant) by a user in the training corpus [23,24,29]. The idea is that if a feature occurs with a high probability in the relevant documents but with a low probability in the non-relevant documents, then this feature is a good indicator of relevance and should be assigned a high weight in the profile. Conversely, if such a feature occurs with a low probability in the relevant docu-

¹ We will use push and routing interchangeably.

ments but with a high probability in the non-relevant documents, then this feature is not a good indicator of relevance and should be assigned a low weight in the profile or should not even be included in the profile.

Our contribution in this paper is many-fold. First, a major contribution is that we have introduced a new method for profile generation based on the vector space model in Information Retrieval (IR). We have systematically studied and compared the proposed new method with well known existing profiling methods across various routing/matching systems, and have found good results. Second, we unify existing well-known profiling methods with statistical contingency theory. Such unification would ease profile methods implementation in computer programs. Third, we reinforce the importance of good ranking/matching function for the routing performance, i.e. we show that even if we use the best profile, the ranking function still makes a lot of difference in terms of performance.

The rest of the paper is organized as follows. Section 2 summarizes the various profile generation methods and their mathematical representations. It also introduces a new method for profile generation. Section 3 reviews three well-known matching functions to match incoming documents against profiles. Section 4 brings up the research questions and corresponding hypotheses. Section 5 describes the experimental setup for data, performance measures and procedures. The results of these experiments are discussed in Section 6. Section 7 offers conclusions and directions for future research.

2. Approaches to profile generation

Information routing/filtering has been studied by computer scientists and information scientists for a long time. Basically, there are two approaches commonly used in the user modeling for profile learning: one is the traditional information retrieval approach using relevance feedback; the other is the machine learning approach using various feature selection methods. The approaches reviewed in this section can all be applied independently of the matching functions (described in the next section). The discussions throughout the study will be based

Table 1
Contingency table for word j

	Relevant	Non-relevant	
Word $j=1$	A	B	$A+B$
Word $j=0$	C	D	$C+D$
	$A+C$	$B+D$	N

on the statistical contingency table discussed below. We show in this section that all of the profile generation methods examined in this paper can be unified using a simple notation based on contingency table.²

Before we discuss the various approaches to selecting words to include in the user's profile, we briefly review some of the prerequisite background on contingency tables, one of the widely used statistical techniques for categorical data analysis.

A contingency table for word j in the training data set is defined in Table 1.

Here A is the number of relevant documents in which word j appears, B is the number of non-relevant documents in which word j appears, C is the number of relevant documents in which word j does not appear, D is the number of non-relevant documents in which word j does not appear. N is defined to be $= (A+B+C+D)$.

Now we proceed to describe the approaches to profile learning and see how these approaches can be cast using the contingency tables just described. As stated earlier the first broad category is the approach based on traditional information retrieval techniques using relevance feedback. In this we will first discuss the Robertson's Selection Value (RSV) method and then introduce our own new method for profile learning. Next we will describe two prominent methods in the second approach of machine learning using feature selection. We have chosen these methods (RSV and the two machine learning ones) to review as they have made substantial theoretical and empirical contributions.

² Our comparative study will not include the classic Rocchio relevance feedback framework [24] because of its dependence on matching function and inferior performance in some evaluation studies [26].

2.1. Traditional information retrieval approach using relevance feedback

2.1.1. Robertson's Selection Value

Robertson [20,21] devised a term selection method based on the probability theory. Robertson Selection Value (RSV) is defined as follows:

$$RSV = (p_i - q_i)RW_i \quad (1)$$

where $p_i = P(w_i = 1 | R)$ is the probability of the presence of word i given that a document is relevant, and $q_i = P(w_i = 1 | \bar{R})$, which is the probability of the presence of word i given that a document is non-relevant. Relevance weights for word i , RW_i , is calculated by

$$RW_i = \log \frac{p_i(1 - q_i)}{q_i(1 - p_i)} \quad (2)$$

RW_i can be obtained based on the probabilistic binary independent model and Bayesian decision rule (Refs. [21,22,28]). According to Ref. [21], q_i can be safely ignored in Eq. (1) since q_i is either much smaller than p_i or RW_i is much smaller for large q_i . So the final RSV formula can be simplified as:

$$RSV = p_i RW_i = p_i \log \frac{p_i(1 - q_i)}{q_i(1 - p_i)} \quad (3)$$

Considering the convention of contingency table as shown in Table 1, $p_i = A/(A + C)$ and $q_i = B/(B + D)$, and ignoring $A + C$, which is a constant for all words, RSV can be simplified to:

$$RSV = A \log \frac{A \times D}{B \times C} \quad (4)$$

All the words from relevant documents are sorted in descending order by their RSV value. For a given profile size (say 200) the top 200 words along with the RSV values as their weights are included in the user profile for routing experiments. The profile size here refers to the number of words (along with their weights) that need to be included to best describe the user's preferences.

2.1.2. Document and Relevance Correlation (DRC)

Inspired by the RSV method discussed above and the cosine similarity measure in the Vector Space model [25], we designed a new profile generation method called the DRC method.

Suppose we have the following indicator function

$$I_{ij} = \begin{cases} 1 & \text{if word } j \text{ is in document } i \\ 0 & \text{otherwise} \end{cases}$$

For a given word j , its binary distribution in all training data can be represented using the following vector (assume we have only six documents):

$$I_j = \langle 1 \ 1 \ 0 \ 0 \ 1 \ 1 \rangle \quad (5)$$

Now suppose we have the following relevance information about the six documents

$$Rel = \langle 1 \ 0 \ 0 \ 0 \ 1 \ 0 \rangle \quad (6)$$

where 0 means non-relevant and 1 means relevant. Then the similarity between word j 's binary distribution in training data and the relevance judgment can be calculated following the Cosine similarity measure as follows:

$$S = \frac{1 + 1}{\sqrt{1 + 1 + 1 + 1} \times \sqrt{1 + 1}} = 0.702 \quad (7)$$

Mathematically, the above calculation can be generally represented as

$$RCV_j = \frac{\sum_{i=1}^N I_{ij} Rel_i}{\sqrt{\sum_{i=1}^N I_{ij}^2} \sqrt{\sum_{i=1}^N Rel_i^2}} \quad (8)$$

We call this new criteria *Relevance Correlation Value* (RCV). Here $j \in 1; 2; N_M$, where N_M is the total number of unique words in the relevant training documents. I_{ij} is an indicator function as defined

above. Rel_i is the relevance judgment (0 or 1) for document i .

Because I_{ij} is either 1 or 0, and if we follow the conventional contingency table as shown in Table 1 and drop the subscript j for simplification, then Eq. (8) can be simplified to:

$$RCV = \frac{A}{\sqrt{A+B}\sqrt{A+C}} \quad (9)$$

Following the design of RSV defined in Eq. (3), DRC is defined to be

$$DRC = p_i RCV \quad (10)$$

where $p_i = P(w_i = 1 | R)$, which is the probability of the presence of word i given that a document is relevant.

Note that the difference between DRC and RSV is that we replace the RW in the RSV formula with RCV in the DRC formula. The rationale is that the cosine measure can also be used to assign relevance weights for document terms based on relevance feedback information.

Making appropriate substitutions and simplification as done previously, Eq. (10) can be reduced to:

$$DRC = \frac{A^2}{\sqrt{A+B}} \quad (11)$$

We call the criteria defined in Eq. (11) as the DRC method. This method combines the information of the word frequency in relevant documents and the association between the word's distribution information with the user's relevance judgments. As with RSV, all the words from the relevant documents are sorted in a decreasing order of DRC and the top ranked (profile size) words form the user profile.

2.2. Machine learning approach using feature selection methods

The representative feature selection methods in machine learning, which have been found successful in various text routing and categorization experiments, include Information Gain used in decision tree learning, Chi-Square test used in categorical attribute independence test, and correlation coefficient, a variation of the Chi-Square test.

2.2.1. Information gain

Information gain is widely used as a feature selection tool in machine learning [12,30]. Features are the attributes in a data set. In our case, they refer to words in the training data set. Information gain is derived based on entropy measure in information theory.

Given a collection S , containing positive (relevant) examples and negative (non-relevant) examples, the entropy of S relative to this Boolean classification is

$$\text{Entropy}(S) = -p\log_2 p - q\log_2 q \quad (12)$$

where p is the proportion of positive examples, q is the proportion of negative examples and $p+q=1$. Entropy measures the impurity of an arbitrary collection of examples. Another interpretation of entropy from information theory is that it specifies the minimum number of bits of information needed to encode the classification (membership) of an arbitrary member of S . For example, if $p=1$, then entropy is zero and no bits are required. If $p=0.5$, then entropy is 1 and 1 bit is required to indicate the membership of a randomly drawn example.

With the entropy defined as above, we can proceed to define information gain (IG). As a measure on the effectiveness of an attribute in classifying training examples, information gain is simply the expected reduction in entropy caused by partitioning the training examples according to this attribute. In other words using information theory, it also measures the number of bits of information obtained for relevance prediction by knowing the presence or absence of a word in a document. The bigger the IG value of a word, the higher the chance that the presence of such a word would help for relevance prediction.

The information gain of a word for relevance prediction is defined as follows:

$$IG(t) = \text{Entropy}(S) - \sum_{v \in \{t, \bar{t}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (13)$$

$$\begin{aligned} IG(t) = & -P(R)\log P(R) - P(\bar{R})\log P(\bar{R}) \\ & + P(t)(P(R|t)\log P(R|t) \\ & + P(\bar{R}|t)\log P(\bar{R}|t)) \\ & + P(\bar{t})(P(R|\bar{t})\log P(R|\bar{t}) \\ & + P(\bar{R}|\bar{t})\log P(\bar{R}|\bar{t})) \end{aligned} \quad (14)$$

where P is the probability function. S_t is the collection of documents containing word t and $S_{\bar{t}}$ is the collection of documents not containing word t . Note that $-P(R)\log P(R) - P(\bar{R})\log P(\bar{R})$ is a constant and is the same for all words in Eq. (13). So the equation can be simplified as follows:

$$\begin{aligned} IG(t) = & P(t)(P(R | t)\log P(R | t) \\ & + P(\bar{R} | t)\log P(\bar{R} | t)) \\ & + P(\bar{t})(P(R | \bar{t})\log P(R | \bar{t}) \\ & + P(\bar{R} | \bar{t})\log P(\bar{R} | \bar{t})) \end{aligned} \quad (15)$$

After making appropriate substitutions based on Table 1, this equation can be reduced to:

$$\begin{aligned} IG(t) \approx & \frac{1}{N} \left(A \log \frac{A}{A+B} + B \log \frac{B}{A+B} \right. \\ & \left. + C \log \frac{C}{C+D} + D \log \frac{D}{C+D} \right) \end{aligned} \quad (16)$$

The procedure of profile construction using information gain is similar to that of the previous two methods. For each unique word in the relevant training documents, we compute the information gain for that word. All the words are arranged in the decreasing value of IG and the top ranked (based on profile length) words form the user profile.

2.2.2. Correlation coefficient

Correlation coefficient \mathcal{C} was first proposed by Ng et al. [14] as a replacement of χ^2 as a feature selection method for information retrieval experiments. Recall that in classical statistics, the χ^2 is used to measure the independence between two categorical variables, which are word t and relevance R in our case. Higher Chi-Square values above the upper-tailed critical values (3.841 for $\alpha=0.05$) means there is a relationship between the word t and relevance R . Based on the definition in Ref. [14] and using Table 1, the χ^2 can be reduced to:

$$\chi^2 = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \quad (17)$$

Though χ^2 has been widely used in machine learning for feature selection, recently it has been found that χ^2 not only selects words that are indicative of relevance of a document, but also those that are indicative of non-relevance of a document as well. The empirical study by Ng et al. [14] stated clearly this side effect. As a remedy for this side effect, they proposed a variation of χ^2 as defined in Eq. (17), called correlation coefficient \mathcal{C} , as follows:

$$\mathcal{C} = \frac{\sqrt{N} \times (AD - CB)}{\sqrt{(A + C) \times (B + D) \times (A + B) \times (C + D)}} \quad (18)$$

It is not difficult to see that $\mathcal{C}^2 = \chi^2$. So \mathcal{C} can be viewed as a “one-sided” χ^2 metric [14]. Ng et al. claimed that the new \mathcal{C} can help pick up those words that only come from relevant documents and are indicative of relevance of a document. Now since $A + C$ and $B + D$ are constants for all words, they are omitted from calculation. Eq. (18) can be simplified as

$$\mathcal{C} = \frac{\sqrt{N} \times (AD - CB)}{\sqrt{(A + B) \times (C + D)}} \quad (19)$$

Similar to the procedure of building profile using information gain, we can calculate the \mathcal{C} value for every unique word in relevant documents and only the top few words (the profile size) can be selected to represent the user profile.

There are also other related research areas in user modeling and personal browser/search agent [1,11,13,18]. These approaches are often modifications of the above mentioned research in terms of profile representation.

In summary, the above reviewed methods (except DRC, which is our own design) are the dominant profile construction methods in IR. Shutze et al. [26] did a limited comparison between Chi-Square and Rocchio relevance feedback framework. No existing routing or push literature can be found to help vendors decide which method works best among methods examined in this paper. We seek to bridge the gap by doing a comparative study of these methods.

3. Approaches to match incoming documents with profiles

In order to systematically test the efficacy of the four profile generation methods discussed in Section 2, three well-known (considered universally good) routing systems are deployed to perform this cross-system empirical study: Okapi system [23], INQUERY system [29] and SMART system [27]. These systems differ mainly on the matching functions. The matching functions utilized by these three systems are summarized as follows:

1. Okapi BM25 [23]

$$\text{Sim}_O(P, D) = \sum_{T \in P} \frac{3 \times \text{tf}}{0.5 + 1.5 \times \frac{\text{length}}{\text{length}_{\text{avg}}} + \text{tf}} \times \log \frac{N - \text{df} + 0.5}{\text{df} + 0.5} \times \text{QTW} \quad (20)$$

2. Pivoted TFIDF [27]

$$\text{Sim}_P(P, D) = \sum_{T \in P} \frac{1 + \log(\text{tf})}{1 + \log(\text{tf}_{\text{avg}})} \times \log \left(\frac{N + 1}{\text{df}} \right) \times \frac{1}{0.8 + 0.2 \times \frac{\text{length}}{\text{length}_{\text{avg}}}} \times \text{QTW} \quad (21)$$

3. INQUERY [29]

$$\begin{aligned} \text{Sim}_I(P, D) &= \sum_{T \in P} 0.4 + 0.6 \\ &\times \left(0.4 \times H + 0.6 \times \frac{\log(\text{tf} + 0.5)}{\log(\text{tf}_{\text{max}} + 1.0)} \right) \\ &\times \frac{\log \left(\frac{N}{\text{df}} \right)}{\log(N)} \times \text{QTW} \end{aligned} \quad (22)$$

where tf is the term frequency of a term (word) in the document text; QTW is the term weighting strategy of

a term (word) in the query text. tf is commonly used as the default weighting strategy. N is the total number of documents in the collection; df is the number of documents in the collection in which the term under consideration is present; ‘length’ is the length of the document (in words); $\text{length}_{\text{avg}}$ is the average document length in the collection (in words); tf_{avg} is the average term frequency of all the terms in the collection; tf_{max} is the maximum term frequency of all the terms in the collection; $H = 1.0$ if $\text{tf}_{\text{max}} \leq 25$, $25/\text{tf}_{\text{max}}$ otherwise.

4. Research questions

We seek to answer the following questions in this paper:

- (1) Which profile method performs better for the same individual matching function?
- (2) Is the answer in (1) generalizable to different matching functions? That is, is there such a profile generation method that gives the optimal performances for all matching functions?
- (3) What is the best way to assign weights for user profile terms?

Various studies [14,15,20,22] suggest that the weight assignment for profile terms is very important for the routing performance. Two obvious ways of assigning weights will be studied: the simple term frequency of terms in user-provided topics/queries or the obtained weights from the various profiling method formulae.

- (4) Should a user’s profile contain a fixed number of terms or varying number of terms?

In TREC conference [9], several participants use a fixed number of term representation scheme for profile representation, which is also adopted in other comparative studies [26]. The fixed number scheme is obviously a simplification of the profile representation task to reduce the learning overhead involved in large-scale applications. It is of interest to know and test empirically what is the performance penalty using this scheme and would this penalty warrant its wide application for profile representation.

5. Experimental setup

5.1. Data

Following the suggestion of the TREC 7 routing track [9], we use the text corpus from Associate Press, which consists of 3 years (1988–1990) news-wire [9]. The AP news-wire covers a broad variety of domains and the documents average roughly 450 words in length. A sample document is shown in Fig. 1.

We use AP88 data set (79,919 documents) as the training data to learn the profiles, AP89 and AP90 (162,999 documents) data sets as the test data to test the efficacy of these profile generation methods and perform cross-system comparison. These data sets were chosen as they follow the time sequence, i.e. AP88 is for the year 1988, AP89, and AP90 are for the years 1989, and 1990, respectively. Hence training is done on 1988 data, while testing is done on the 1989 and 1990 data.

There are a total of 50 different user-provided topics/queries used in the TREC 7 routing experiment. One sample topic is shown in Fig. 2. All the 242,918 documents in the AP collection were judged for relevance by experts for these 50 topics.

A total of 42 (out of 50) topics are chosen for the profile learning experiment because there are less

than 4 (6 of 8 even have 0 relevant documents for training) relevant documents available for training for the remaining 8 (out of 50) topics. Such a small number of relevant documents in these eight topics will not help profile learning method to learn representative profiles. So all results reported in profile learning studies are for those remaining 42 topics, each of which has more than 3 (≥ 4) relevant documents in the training data. We leave the examination of effect of very few relevant documents for future studies. A possible (although approximate) way around this problem of very few relevant documents is to include semantically similar documents in the relevant set.

These 42 topics are the original questions raised from users to represent their interests and thus they can be used to generate explicit profiles as discussed in Section 1. We are going to treat these explicit profiles as baselines and compare their performance with those implicit profiles constructed by the 4 profile generation methods (RSV, IG, DRC, and CC) discussed in Section 2.

Two different versions of these 42 topics were indexed. In one version, only the title portion of the topics is indexed. These profiles are called *short user-provided profiles* or *short explicit profiles*. This is to simulate a scenario in which the query length is

```
<DOC>
<DOCNO> AP881223-0040 </DOCNO>
<FILEID>AP-NR-12-23-88 0401EST</FILEID>
<FIRST>r i PM-Obit-Suroi 12-23 0160</FIRST>
<SECOND>PM-Obit-Suroi,0165</SECOND>
<HEAD>Yugoslav Ambassador To Madrid Dies In Car Accident</HEAD>
<DATELINE> MADRID, Spain (AP) </DATELINE>
<TEXT>
Redzai Suroi was 59. Suroi died
Thursday near Guadalajara as he returned alone to Madrid from a
private trip, said embassy counselor Zoran Raicevic. Suroi, a native
of Erizren in the autonomous province of Kosovo, had held the post of
adjunct to the Yugoslav foreign minister before coming to Spain in
October 1985. He also served as ambassador to Mexico from 1978 to
1982 and to Bolivia from 1970 to 1974, Raicevic said. Suroi, a law
graduate of Belgrade University, worked as a journalist for 15 years,
and became director of Radio Pristina in Kosovo before beginning his
diplomatic career in 1970, Raicevic said. Survivors include his wife,
a son and a daughter, the counselor said.
</TEXT>
</DOC>
<DOC>
```

Fig. 1. Sample document from AP news-wire collection.

```

<top>
<head> Tipster Topic Description
<num> Number: 002
<dom> Domain: International Economics
<title> Topic: Acquisitions
<desc> Description:
Document discusses a currently proposed acquisition involving a U.S.
company and a foreign company.
<narr> Narrative:
To be relevant, a document must discuss a currently proposed
acquisition (which may or may not be identified by type, e.g., merger,
buyout, leveraged buyout, hostile takeover, friendly acquisition).
The suitor and target must be identified by name; the nationality of
one of the companies must be identified as U.S. and the nationality of
the other company must be identified as NOT U.S.
<con> Concept(s):
1. acquisition, takeover
2. suitor, target
3. merger, buyout, leveraged buyout (LBO)
4. arb, arbitrage, arbitrage, leverage, offer, bid, tender, purchase
5. anti-takover, poison pill, white knight, restructure, sale of
assets, recapitalization
</top>

```

Fig. 2. Sample topic description in TREC 7.

very short [10] possibly due to users unwillingness to provide a lengthy description about his/her information needs. The number of title terms contained in the 42 short user-provided profiles varies from 4 to 18. In the other version, we indexed all the components of the 42 topics (title, description, narrative, and concepts). Correspondingly, these longer profiles are called *long user-provided profiles* or *long explicit profiles*. The number of terms for 42 long user-provided profiles varies from 17 to 96. This represents those situations when a user is willing to provide a lengthy description about his/her information need.

The topics and the relevant document distribution information for the experiment is summarized in Table 2.

5.2. Performance measure

Following the convention of TREC 7 routing system evaluation procedures, the performance of various routing systems is also evaluated using the standard performance measure called *Average Precision* (P_{avg}), which is calculated as follows:

The average of precision score is calculated every time a relevant document is found, normalized by

total number of relevant documents in the entire collection. Mathematically it can be expressed as:

$$P_{avg} = \sum_{i=1}^{TRel} P_i / TRel, \quad \text{where } P_i = i / \text{Rank}_i \quad (23)$$

Here $TRel$ is the total number of relevant documents for a given query, and Rank_i is the ranking position for the i th relevant document. All results reported in the following result section are based on P_{avg} .

5.3. Experimental design

As mentioned above, the purpose of our experiments is to empirically compare different profile learning approaches and see their performance in real text routing experiments under various control conditions. We follow a full factorial design to test for statistical significance. The configuration of the experiment is summarized in Table 3.

As shown in Table 3, there are four main factors in our study: profile, weight, system and size. Taking into account various levels for each factor, there are total $4 \times 2 \times 3 \times 20 = 480$ configura-

Table 2
Data set used for routing experiments

Topic number	Number of relevant documents (Training)	Number of relevant documents (Test)
1	122	344
2	260	459
3	60	220
4	30	94
5	28	68
6	71	270
7	116	311
8	28	66
9	28	107
10	85	265
11	125	427
12	180	624
13	8	86
14	43	116
15	59	107
16	28	142
17	89	224
18	62	130
19	59	269
20	47	158
21	21	29
22	395	1238
23	77	237
24	111	300
25	23	65
26	7	61
27	7	20
28	7	66
29	4	7
36	4	10
38	43	276
40	74	236
41	12	74
42	66	151
43	25	102
44	28	152
45	51	52
46	26	74
47	26	89
48	6	30
49	15	55
50	5	6
Sum.	2561	7817
Avg.	61	186

rations of experimental conditions. For each configuration of experimental condition, all 42 queries are run and results are recorded for data analysis. So in total there are 21,060 data points collected for analysis.

The description of each factor follows. Four different profile learning methods (RSV, IG, DRC, and CC), as discussed in Section 2, are used. Three different routing/matching functions (Okapi BM25, INQUERY, and SMART Ptfidf) as discussed in Section 3 are used. *Profile Weighting* means how the weights are assigned for each term in the profile. There are 2 ways of assigning term weights for profile terms: Query Term Frequency (QTF) and Obtained Weights (OW) [15,20]. QTF means that the original term frequency from the user-provided profile is used to assign weights for profile terms. If the profile term in the learned profile is from the relevant documents and is not provided by the user, its QTF is assigned 1. A second way to assign weights to the profile terms is to use the relevance weights obtained by the various term selection formula discussed in Section 2. These weights are called OW in Table 3. The factor P_size refers to the size of the profile, i.e. how many terms are there in each profile. For experimental purpose this is varied from 10 to 200 in increments of 10. A very low value means the profile is very selective while a high value means the profile is very general. A selective profile is desired for a narrow focused search and retrieval, while a more general profile is desired when the search can be broad but still within the overall context of the subject topic.

In order to answer the research questions raised above, the performance of user-provided profiles (explicit profiles) is used as a baseline for benchmark comparison with that of implicit profiles learned by best profile methods for each routing system. These best profile generation methods are identified based on the performance results of various profiling methods on the training data for each routing system. Since there are three routing systems used: Okapi, SMART

Table 3
Profile learning experimental setup

Factor label	Meaning	Levels	Values
Profile	Profiling method	4	RSV, IG, DRC and CC
Weight	Profile weighting	2	Query Term Frequency (QTF), Obtained Weights (OW)
System	Matching function	3	Okapi, INQUERY, SMART
p_size	Profile size	20	10, 20, 30, ..., 200

and INQUERY, three best profiling methods will be identified (one for each routing system). A final comparison of these three profiling methods will help answer the research questions.

6. Results and discussions

All experiments were run on an IBM Linux server. All programs were coded in C. Analysis of variance (ANOVA) statistical analysis was applied to the experimental data to help answer some of the research questions.

6.1. ANOVA analysis

The model is set up as follows (in SAS syntax):

$$\text{Full Model: } p_avg = \text{system} | \text{profile} | \text{weight} | p_size \quad (24)$$

Basically, we use p_avg as the dependent variable and those main factors and their full factorial interaction are used as independent variables.

The results of ANOVA analysis are summarized in Table 4.

As can be easily seen from Table 4, all the main factors—system, profile, weight and p_size —have significant impacts on the overall routing performance results ($p < 0.0001$). However, because most of the higher-order interactions are significant ($p < 0.05$), these main effects really do not convey any more meaningful information since these high-order interactions suggests that the performance under one factor is also under the influence of another factor and these influences do not work consistently [19].

We begin with the highest interaction term. The four-way interaction between these four factors are non-significant ($F = 0.10$, $p (= 1.0000) > 0.05$), suggesting that such four-way interaction among these factors does not exist overall. All the three-way interactions are not significant except for profile \times weight \times system, which suggests that selecting appropriate profiling methods along with the weighting strategies will greatly impact the routing systems' performance. Profile size (p_size), though, does not

Table 4
ANOVA analysis results

Source	<i>df</i>	<i>F</i> value	Pr> <i>F</i>
System	2	597.33	<0.0001
Profile	3	410.29	<0.0001
Weight	1	17.08	<0.0001
p_size	19	9.27	<0.0001
Profile \times system	6	57.23	<0.0001
Weight \times system	2	15.88	<0.0001
Profile \times weight	3	16.18	<0.0001
System \times p_size	38	7.73	<0.0001
Profile \times p_size	57	2.12	<0.0001
Weight \times p_size	19	1.24	0.2176
Profile \times weight \times system	6	4.31	0.0002
Profile \times system \times p_size	114	0.72	0.9902
Weight \times system \times p_size	38	0.47	0.9975
Profile \times weight \times p_size	57	0.64	0.9835
Profile \times weight \times system \times p_size	114	0.10	1.0000

have interaction impacts with the weighting strategies used ($p = 0.2176 > 0.05$), it does impact the performances for different profiles ($p(\text{profile} \times p_size) < 0.0001$) and for different systems ($p(\text{system} \times p_size) < 0.0001$).

The strong statistical significance of the main effects and those interaction effects of the main factors again statistically warrants the main argument of our research: the selection of appropriate profiling methods is a key factor for the success of a routing system. However, it is not clear from the above analysis how the scales of these impacts look like in real routing performance. We will next look at these issues in more detail by performing graphical and statistical analysis.

6.2. A look from the perspective of profiling methods

The performance of the four profile generation methods under different experimental conditions is summarized in Figs. 3–6. In each figure, each point represents the performance result measured by P_avg , averaged over all 42 profiles. These results are reported for various profile sizes by using different combinations of routing systems and profile weighting strategies.

As can be seen from these four figures, various profile generation methods indeed produce different performance results for routing experiments. The magnitude of these impacts can be easily seen after

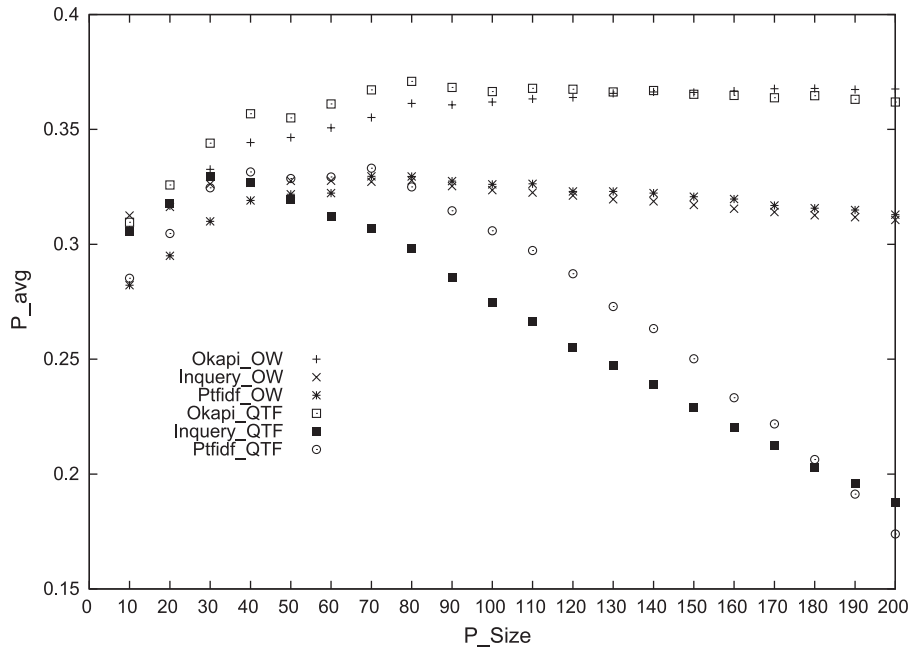


Fig. 3. Effects of RSV variations on routing performance measured by P_avg.

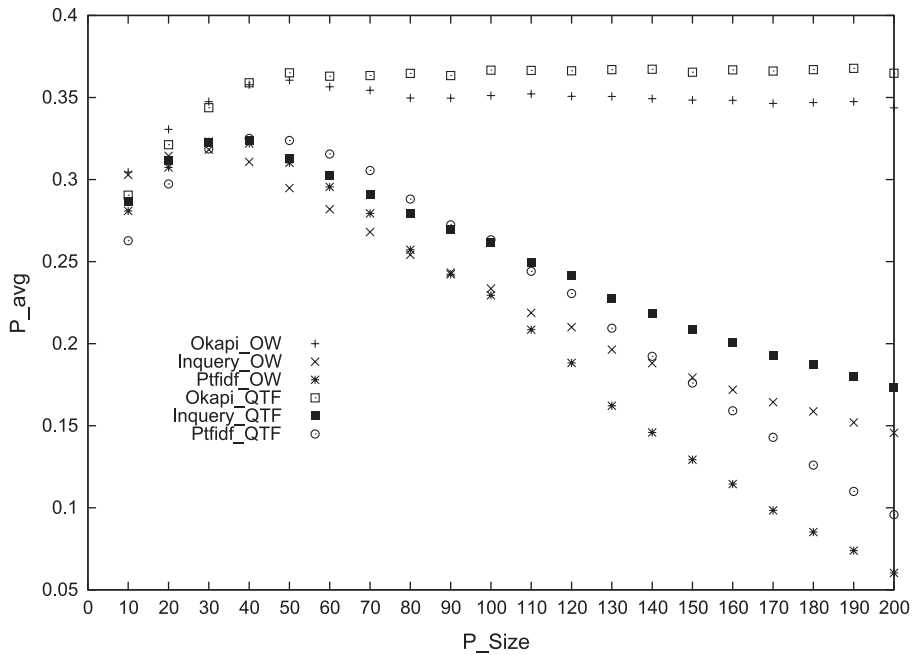


Fig. 4. Effects of IG variations on routing performance measured by P_avg.

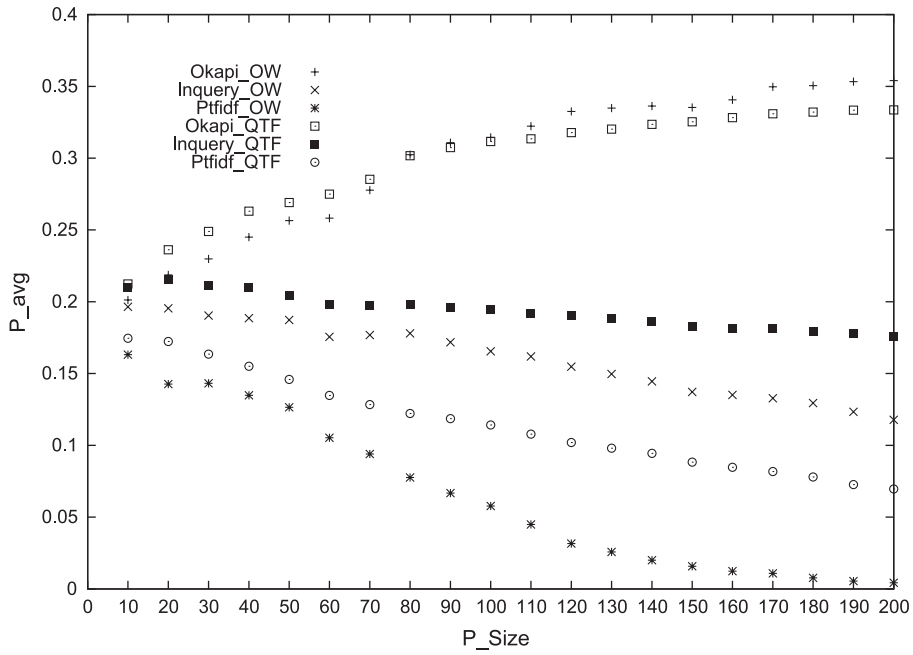


Fig. 5. Effects of CC variations on routing performance measured by P_avg.

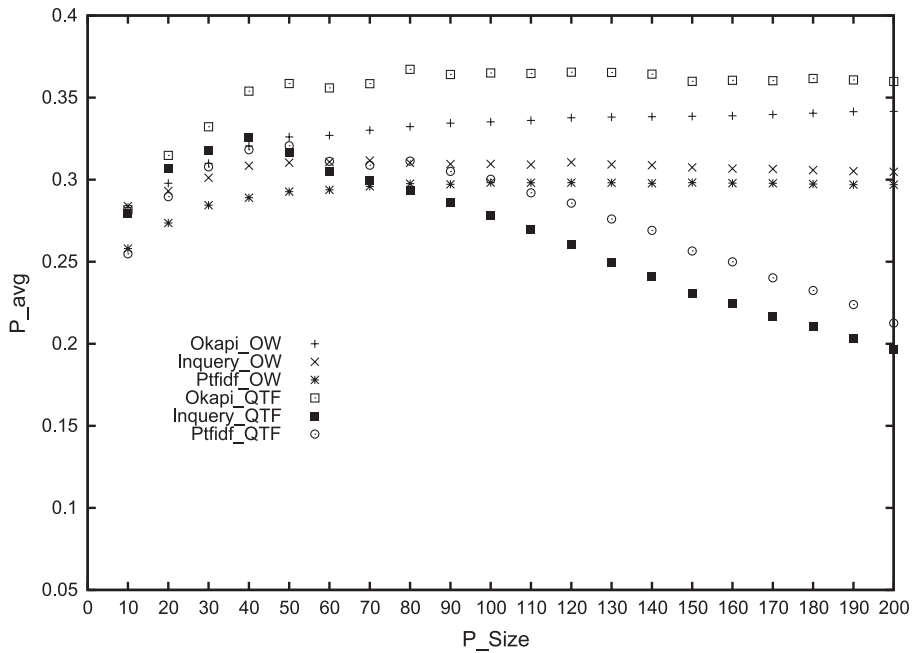


Fig. 6. Effects of DRC variations on routing performance measured by P_avg.

examining these diagrams. There are several observations that can be made from these four figures:

- (a) One thing that is consistent in these four figures is that all profiling methods work best when combined with Okapi system (Multiple comparison tests (Tukey) also show that these differences between Okapi and other systems for the same profiling methods are all statistically significant (p 's < 0.001)). This can be clearly verified by the dramatic performance gap between Okapi with other routing systems. This result concurs with past TREC results and re-confirms that Okapi BM25 matching function certainly has a huge advantage in routing experiments.
- (b) RSV, IG, and DRC profiling methods when used with Okapi system show that the performance is not affected by the number of terms in a profile when the number reaches a certain value (range of 60 to 80). CC seems to be very different from other profile generation methods in that it requires more terms in its profiles to gain in performance. As can be seen from Fig. 5, as the number of terms increase in user profiles, its performance in Okapi system also improves and stabilizes when the number of terms is near 190.
- (c) One point that deserves further attention here is the performance of the DRC profiling method, which is our own design. When QTF is used to assign values for profile terms, it has similar performance as compared with RSV and IG profiling methods when Okapi system is used. A follow-up t -test does not show any statistical difference. But when OW is the default profile value assignment method, its performance on all three routing systems is more stable (the least variance) than any other profile generation method when the number of terms in profile is above a threshold. Its performance is very similar to RSV in overall. IG seems to have the biggest performance drop when used in routing system other than Okapi.
- (d) It can be seen from these figures that the P_{avg} value increases initially, remains steady for some time, and then decreases with increasing values of P_{size} . The decrease in performance with increasing profile size may seem counter-intuitive. The reason why this happens is that when profile

size is too large then there is higher likelihood of introducing noise words in the profile just to maintain the profile size. Presence of such noise words eventually leads to decreasing performance. Later in the paper we examine if there is a performance penalty to have a fixed profile size.

6.3. An examination from the perspective of the routing systems

Figs. 3–6 look at the performance comparisons for each profiling method. Another angle to look at the performance comparisons is from the routing systems perspective. This type of analysis will have more practical implications since it will help us understand better the practical impacts of different profiling methods given a routing system and help us better choose the best profiling method. The results of all profiling methods for each routing system are illustrated in Figs. 7–9.

For Okapi system using the BM 25 formula, RSV QTF is the best overall profiling method when the number of terms in a profile varies from 10 to 200, as shown in Fig. 7. However, the differences between RSV QTF, IG QTF, and DRC QTF are not significant (multiple comparison tests show that all pair-wise comparisons have $p = 1.000 > 0.05$). CC profiling method is the worst method among the four examined, which is true especially when a small number of terms (< 90) are included in the profiles. As the number of terms included in the profiles increases, the performance gap between CC and other profiling methods decreases accordingly. Overall, RSV, IG and DRC, combined with QTF, do not see any performance gain as the number of terms in profiles increases.

INQUERY and SMART seem to display very similar patterns if we examine the performance results in Figs. 8 and 9. Both figures show a falling trend as the number of terms in profiles increase, which is quite different from that of Okapi. RSV, IG and DRC, when combined with QTF, perform well when the number of terms in a profile is small (30 to 50). However, their performance begins to fall off when the number of terms in profiles increases. The OW term weight assignment seems to work well for larger profiles. DRC OW performs very consistently even with the variation of profile size, though its performance may not be the best among all methods.

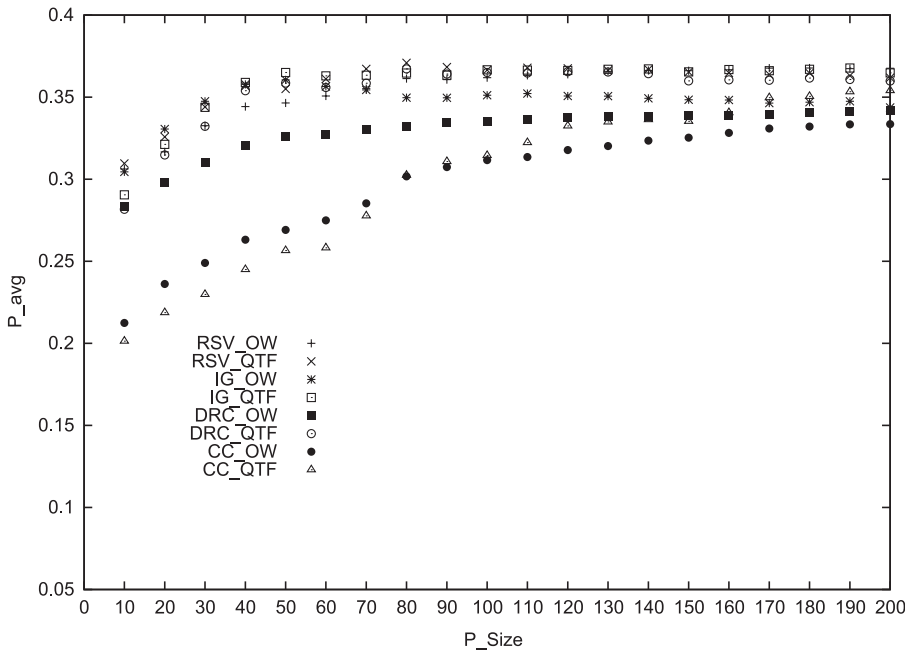


Fig. 7. Effects of profiling methods on Okapi routing performance measured by P_avg.

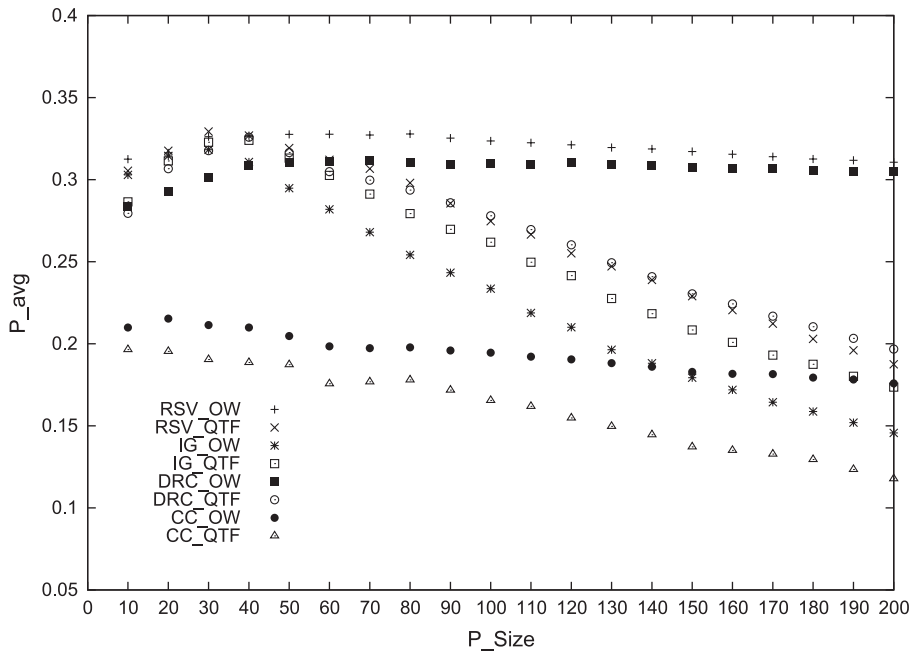


Fig. 8. Effects of profiling methods on INQUERY routing performance measured by P_avg.

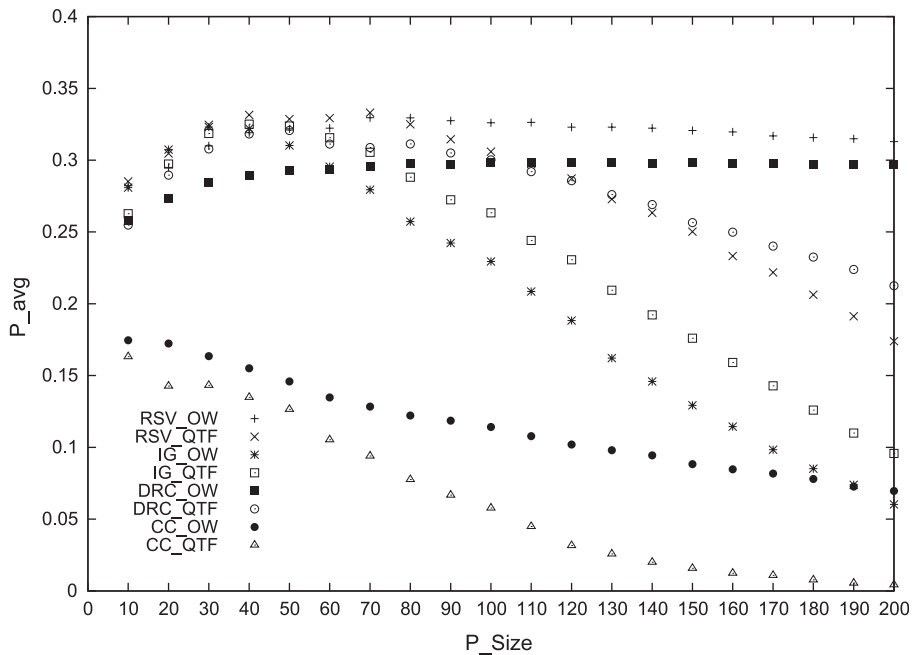


Fig. 9. Effects of profiling methods on SMART routing performance measured by P_avg.

Overall, the best performance results are still obtained by RSV QTF with profiles of very small size (30 to 40) for both INQUERY and SMART matching functions.

6.4. Benchmark comparison with the baseline

In order to see more clearly how the four implicit profile generation methods fare against each other and the baseline using user-provided long query (explicit profile), numerical results on these perfor-

mance comparisons are summarized in Table 5. In this table, first column is the routing systems used in our experiments, where O stands for Okapi, I for INQUERY, and S for SMART. In each cell of the table starting column 3, three values are reported. The first value is the maximum P_avg averaged over 42 queries among 20 different configurations (P_avg from 10 to 200) of profile representation. The second value is the performance gain over baseline (reported in column 2 of the table) in percentage under the same routing system. The third value is the number

Table 5

Summary of the best performing results for various profile generation methods and their comparison with the baseline

	Baseline	RSV		IG		CC		DRC	
	Long	QTF	OW	QTF	OW	QTF	OW	QTF	OW
O	0.32	0.3709	0.3678	0.3678	0.3605	0.354	0.3336	0.3672	0.3416
	-	15.91%	14.94%	14.94%	12.66%	10.62%	4.25%	14.75%	6.75%
	-	80	180	190	50	200	200	80	200
I	0.2883	0.3293	0.3278	0.324	0.3182	0.1965	0.2153	0.3259	0.3116
	-	14.22%	13.70%	12.38%	10.37%	-31.84%	-25.32%	13.04%	8.08%
	-	30	80	40	30	10	20	40	70
S	0.269	0.3331	0.3296	0.3251	0.3231	0.1632	0.1746	0.3207	0.2982
	-	23.83%	22.53%	20.86%	20.11%	-39.33%	-35.09%	19.22%	10.86%
	-	70	70	40	30	10	10	50	150

All results are statistically significant (*t*-test with $p < 0.005$).

of terms in the user profiles that obtain the best performance value. This table clearly shows the advantage of RSV over other profile generation methods. For example, RSV+QTF and RSV+OW perform almost equally well for all three routing systems examined. This is consistent with the conclusions we got earlier. However, this conclusion is not applicable to other methods. When QTF is used, both RSV and DRC require a small number of terms in profiles to gain optimal performance. Among the methods examined, CC performs the worst across the board. Its performance gains over the baselines are even negative in INQUERY and SMART. Overall, RSV+QTF is the best profile generation method in all three routing systems.

6.5. Aggregation results

One result that appears to emerge from Table 5 is that QTF appears to be better than OW for all the combinations of profile generation methods (except CC) and routing systems. To take a closer look at this issue, an aggregated result for all these profile methods across systems are summarized in Table 6 using results from Table 5. As can be seen from this table, QTF clearly has advantages over OW for DRC method, with 0.02 difference. However, the edge of QTF over OW is not very significant for RSV and IG. So if we leave out CC from our discussion, we can say that QTF term weighting strategy is better overall than OW term weighting strategy.

Another aggregation across profile generation methods for all three routing systems is done to see which system gives the best performance. The results are shown in Table 7. The column “Average” is obtained by averaging results in Table 5 by row. Column “Maximum” is also derived from Table 5 and corresponds to the maximum performance results among all profile methods for each row.

Table 6
Aggregated cross-system results for different profile generation methods

RSV		IG		CC		DRC	
QTF	OW	QTF	OW	QTF	OW	QTF	OW
0.3444	0.3417	0.3389	0.3339	0.2379	0.2411	0.3379	0.3171

Table 7
Aggregated results (in P_avg) across profile generation methods for different routing systems

System	Average	Maximum	Method used
Okapi	0.3579	0.3709	RSV_QTF
INQUERY	0.2936	0.3293	RSV_QTF
SMART	0.2835	0.3331	RSV_QTF

“Method Used” is the profile method used to obtain the maximum performance in column “Maximum”. An obvious conclusion that can be made from this table is that Okapi is superior to other routing systems in terms of both average and maximum performance and RSV+QTF is the overall best profile generation method. A follow-up multiple comparison test with Tukey grouping on system factor has shown that the difference between the performance figures of the routing systems is statistically significant with Okapi being the best routing system overall.

6.6. The effect of query difficulty

It is clear from above results and discussions that RSV+QTF is the overall best profile generation method. The above analysis is done mainly from a macro level by analyzing the averaged or aggregated results. One of the interesting questions to ask on a micro level is: How is profile learning affected by the query difficulty? It is known in IR and common practice that for some topics/queries it is very hard to find any relevant documents, while for others it may be very easy to locate new relevant documents. We call this a phenomenon of query difficulty. In order to study the query difficulty and its effects on the routing performance, we compare the performance results of RSV+QTF with that of baseline, i.e. long queries provided by the user. For any given query, the query difficulty can be measured by the magnitude of its routing performance measured in P_avg. If P_avg is very high, then it means the query is relatively easy. If P_avg is very low, then it means high query difficulty. The results are illustrated in Fig. 10.

In Fig. 10, each circle represents 1 of the 42 queries. If the circle is above (below) the bisecting line, then performance obtained by profile learning over original user provided profile (explicit profile/query) is better

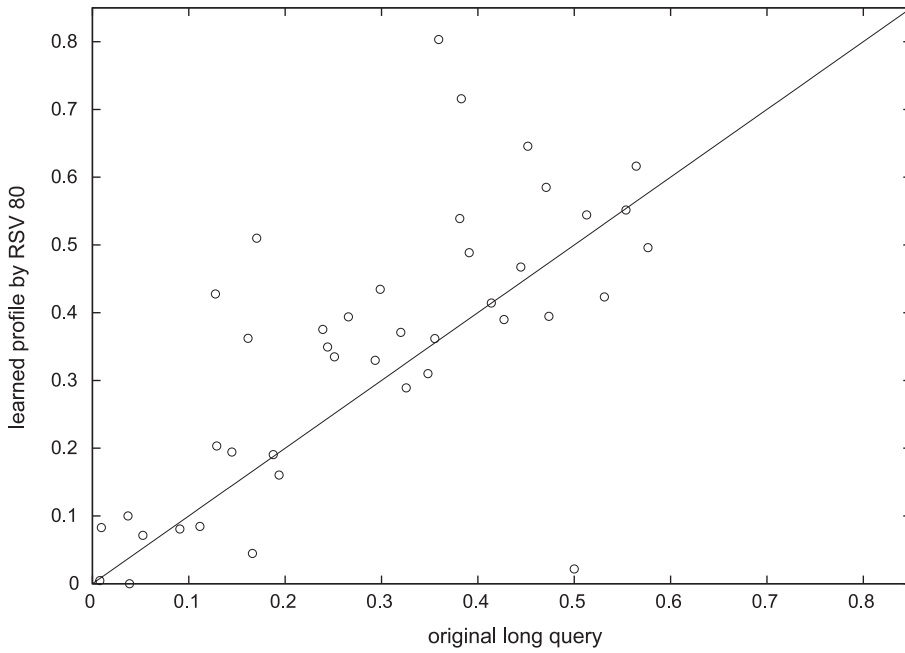


Fig. 10. Performance (P_{avg}) of RSV 80 versus initial query difficulty.

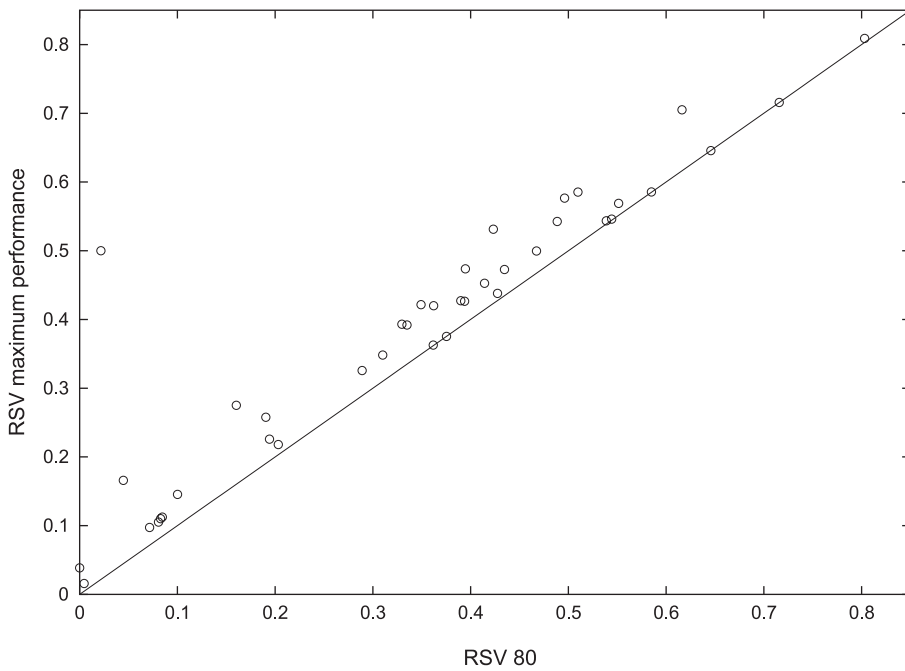


Fig. 11. Maximum performance (P_{avg}) by RSV versus fixed RSV 80.

(worse). The query difficulty decreases as we move away from the origin (0.0, 0.0).

A closer examination of Fig. 10 does not reveal any strong pattern. For difficult queries, which are closer to the origin, profile learning shows some mixed results. As the query difficulty decreases when we move away from the origin, however, we can achieve a significant amount of improvement in performance. This is due to the fact that more relevant training documents are easily available for less difficult queries.

6.7. Fixed number of terms in a profile?

Another interesting question to ask on the micro level is: How does the overall best performing profile generation method fare against manually selected best performance for each individual query? We know from Table 5 that RSV+QTF obtained the best performance with 80 terms selected in user profiles for all 42 queries. Even though RSV+QTF 80 is the best performing method overall, it may not be best for each individual query. For example, there is a high probability that the number of terms included in a profile, which obtains the best performance for this particular query, is not 80. The answer to this question has significant impact for routing experiment strategies. Is it really worthwhile to find the optimal number of terms to be included in each profile? Can we not use a fixed number, like 80, for all profile representation? A comparison between the performance of RSV+QTF 80 and the maximum performances for each individual query is done and the results are summarized in Fig. 11.

Similar interpretation, as in Fig. 10, can be used here. Any circle close to the bisecting line means very close performance. It is not difficult to see that these two maxima obtained in different ways are close to each other, with certain exceptions. This argues for using fixed number of terms in profile representation simply because it is simple and there is not much performance penalty.

7. Conclusions and future research directions

In this paper, we systematically compared profile generation methods and their variations using three

well-known routing systems. The following conclusions can be drawn:

- Implicit profiles generated by profile learning methods are generally superior to explicit profiles specified by users.
- Among the four methods for profile generation, RSV combined with QTF gives the best performance on average for all 42 queries for a given routing system. This result is also generalizable to other routing systems.
- In general, the term weights assigned to profile terms using original term frequency are better than those assigned using the weights calculated from profile generation formula.
- Among the four methods for profile generation, profile generated by RSV is the most robust one. It is not very sensitive to the weight assigning methods: QTF or OW.
- Although overall RSV is the most robust and best performing method, the difference between RSV and IG, and RSV and DRC are not statistically significant when QTF is used as the term weighting strategy for profile terms. In particular, both DRC and IG perform equally well when combined with QTF weighting strategy for terms included in the profiles.
- Given the same profiles, there are still huge statistical performance differences when different routing systems are used. The performance of Okapi BM25 formula is statistically significantly better than that of INQUERY formula and SMART Ptfidf formula for all different profiles.

These conclusions can help push system vendors or designers to better capture and represent users' information needs. We believe the success of the profile representation will be a key step towards improving push(routing) service quality without overloading consumers with non-relevant information.

This work can be extended in several directions.

- One of the conclusions that can be inferred from above discussions is that these profile generation methods select different terms to be included in the profiles. One way of extending this work is to apply the majority-voting strategy or other ensemble

bling techniques [2] to choose terms to be included in user profiles. This deserves more study.

- It is well known in IR literature that matching function has a dramatic impact on the performance of search engines, as well as routing systems as shown above. Okapi formula seems to be very robust in overall performance, but it is still questionable whether Okapi BM 25 is the single best ranking function for each individual user profile. One interesting question to study is to see whether profile learning methods, such as RSV with QTF, can be complemented by other matching function adaptation algorithms, like GA [17] and GP [4–6] to further improve the routing performance. We leave this for future research.
- We can explore the effect of type of retrieval task. Some tasks might be more difficult to retrieve than others because of the involved structure and semantics.
- It would be interesting to see how to combine the implicit profiles with the explicit profiles that the customer has. Such a combination would hopefully lead to enhanced routing performance. One way this could be done is to map categories from both implicit and explicit profiles and elevate the weight associated with such overlapped words.
- In this paper we have utilized well-known routing functions. Future research can attempt to explore how other techniques like natural language processing, name entity extraction, and information extraction could be applied in the context of profile generation and maintenance.

References

- [1] L. Chen, K. Sycara, Webmate: a personal agent for browsing and searching, in: K.P. Sycara, M. Wooldridge (Eds.), *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, ACM Press, New York, 1998, pp. 132–139.
- [2] T. Dietterich, Machine learning research: four directions, *AI Magazine* 18 (4) (1997) 97–135.
- [3] S. Dumais, Using SVMs for text categorization, *IEEE Intelligent Systems* 13 (4) (1998) 21–23.
- [4] W. Fan, M.D. Gordon, P. Pathak, Personalization of search engine services for effective retrieval and knowledge management, *Proceedings of 2000 International Conference on Information Systems (ICIS)*, Brisbane, Australia. AIS, Atlanta, GA, 2000, pp. 20–34.
- [5] W. Fan, M.D. Gordon, P. Pathak, Discovery of context-specific ranking functions for effective information retrieval using genetic programming, *IEEE Transactions on Knowledge and Data Engineering* 16 (4) (2004) 523–527.
- [6] W. Fan, M.D. Gordon, P. Pathak, A generic ranking function discovery framework by genetic programming for information retrieval, *Information Processing & Management* (2003) (in press).
- [7] M. Franklin, S. Zdonik, Data in your face: push technology in perspective, *Proceedings of the 1998 ACM SIGMOD Conference*, ACM Press, New York, NY, 1998, pp. 516–519.
- [8] G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The vocabulary problem in human–system communication, *Communications of the ACM* 30 (11) (1987) 947–971.
- [9] D. Hull, The TREC-7 filtering track: description and analysis, in: E.M. Voorhees, D.K. Harman (Eds.), *Proceedings of the Seventh Text Retrieval Conference*, NIST Special Publication vol. 500-242, 1999, pp. 33–56.
- [10] B.J. Jansen, A. Spink, T. Saracevic, Real life, real users, and real needs: a study and analysis of user queries on the web, *Information Processing & Management* 36 (2) (2000) 207–227.
- [11] F. Menczer, R.K. Belew, Adaptive retrieval agents: internalizing local context and scaling up to the web, *Machine Learning* 39 (2/3) (2000) 203–242.
- [12] T.M. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, 1997.
- [13] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, D. Zabowski, Experience with a learning personal assistant, *Communications of the ACM* 37 (7) (1994) 81–91.
- [14] H. Ng, W. Goh, K.L. Low, Feature selection, perceptron learning, and a usability case study for text categorization, *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM press, New York, NY, 1997, pp. 67–73.
- [15] H.T. Ng, H.T. Ang, W.M. Soon, DSO at TREC-8: a hybrid algorithm for the routing task, in: E. Voorhees, D. Harman (Eds.), *The Eighth Text Retrieval Conference (TREC-8)*, NIST Special Publication, vol. 500-246, 1999, pp. 267–274.
- [16] K.H. Packer, D. Soergel, The importance of SDI for current awareness in fields with severe scatter of information, *Journal of the American Society for Information Science* 30 (3) (1979) 125–135.
- [17] P. Pathak, M. Gordon, W. Fan, Effective information retrieval using genetic algorithms based matching function adaptation. *Proceedings of the 33rd Hawaii International Conference on System Science (HICSS)*, Hawaii, USA, 2000.
- [18] M.J. Pazzani, D. Billsus, Learning and revising user profiles: the identification of interesting web sites, *Machine Learning* 27 (3) (1997) 313–331.
- [19] E.J. Pechazur, L.P. Schmelkin, *Measurement, Design, and Analysis: An Integrated Approach*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.
- [20] S. Robertson, On relevance weight estimation and query expansion, *Journal of Documentation* 42 (1986) 182–188.
- [21] S. Robertson, On term selection for query expansion, *Journal of Documentation* 46 (1990) 359–364.
- [22] S. Robertson, K.S. Jones, Relevance weighting of search

terms, *Journal of the American Society for Information Science*, 27 (1976) 129–146, Reprinted in: P. Willett (Ed.), *Document Retrieval Systems*. Taylor Graham, 1988, pp. 143–160.

- [23] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-4, in: D.K. Harman (Ed.), *Proceedings of the Fourth Text Retrieval Conference*, NIST Special Publication, vol. 500-236, 1996, pp. 73–97.
- [24] J.J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), *The SMART Retrieval System—Experiments in Automatic Document Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [25] G. Salton, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall, New Jersey, 1971.
- [26] H. Schutze, D. Hull, J. Pedersen, A comparison of classifiers and document representations for the routing problem, *Proceedings of the 16th ACM SIGIR'95*, Seattle, USA, ACM Press, New York, NY, 1995, pp. 229–237.
- [27] A. Singhal, G. Salton, M. Mitra, C. Buckley, Document length normalization, *Information Processing & Management* 32 (5) (1996) 619–633.
- [28] C.J. van Rijsbergen, *Information Retrieval*, 2nd edition, Butterworth, London, 1979.
- [29] J. Xu, W. Croft, Query expansion using local and global document analysis, *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, NY, 1996, pp. 4–11.
- [30] Y. Yang, J. Pedersen, A comparative study on feature selection in text categorization, *Proc. 14th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 1997, pp. 412–420.

Weiguo Fan is an assistant professor of information systems and computer science at the Virginia Polytechnic Institute and State University. He received his Ph.D. in Information Systems from the University of Michigan Business School, Ann Arbor, in July 2002. His research interests include personalization, data mining, text/web mining, web computing, business intelligence, digital library, and knowledge sharing and individual learning in online communities. His research has appeared in many prestigious information technology journals such as *Information Processing*

and *Management (IP&M)*, *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, *Information Systems (IS)*, *Decision Support Systems (DSS)*, *ACM Transactions on Internet Technology (TOIT)*, *Journal of the American Society for Information Science and Technology (JASIST)*, *Journal of Classification*, *International Journal of Electronic Business*, and in leading information technology conference such as ICIS, HICSS, AMCIS, WWW, CIKM, DS, ICOTA, etc.

Michael Gordon is a professor of business information technology and associate dean for information technology at the University of Michigan Business School. His research interests include information retrieval, especially adaptive methods and methods that support knowledge sharing among groups; information and communication technology in the service of social enterprise (promoting economic development, providing health care delivery, and improving educational opportunities for the poor); and using information technology along with social methods to support business education. He publishes extensively in leading IT journals such as *Information Processing and Management (IP&M)*, *IEEE Transactions on the Knowledge and Data Engineering (TKDE)*, *Decision Support Systems (DSS)*, *ACM Transactions on Internet Technology (TOIT)*, *Journal of the American Society for Information Science and Technology (JASIST)*, *Information Systems Research*, *Communication of ACM*.

Dr. Praveen Pathak is an Assistant Professor of Decision and Information Sciences at the Warrington College of Business at the University of Florida. He received his Ph.D. in Computer and Information Systems from the University of Michigan Business School, Ann Arbor, in 2000. He also holds a MBA (PGDM) from Indian Institute of Management, Calcutta, and a Engineering degree, B. Tech. (Hons.), from the Indian Institute of Technology, Kharagpur. His research interests include information retrieval, text mining, business intelligence, and knowledge management. His research has appeared in many prestigious journals such as *Decision Support Systems (DSS)*, *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, *Information Processing and Management (IP&M)*, *Journal of the American Society for Information Science and Technology (JASIST)*, and in leading information technology conferences such as ICIS, HICSS, WITS, etc.