Hybrid User Model for Information Retrieval*

Hien Nguyen
University of Wisconsin-Whitewater
Mathematical and Computer Sciences Dept.
800 W. Main Street
Whitewater, WI 53190
nguyenh@uww.edu

Eugene Santos Jr.
Dartmouth College
Thayer School of Engineering
8000 Cummings Hall
Hanover, NH 03755
eugene.santos.jr@dartmouth.edu

Nathan Smith and Aaron Schuett
University of Wisconsin-Whitewater
Mathematical and Computer Sciences Dept.
800 W. Main Street
Whitewater, WI 53190
(smithnd09,schuettai07)@uww.edu

Abstract

In this paper, we report our development of a hybrid user model for improving a user's effectiveness in a search. Specifically, we dynamically capture a user's intent and combine the captured user intent with the elements of an information retrieval system in a decision theoretic framework. Our solution is to identify a set of key attributes describing a user's intent, and determine the interactions among them. Then we build our user model by capturing these attributes, which we call the *IPC* model. We further extend this model to combine the captured user intent with the elements of an information retrieval system in a decision theoretic framework, thus creating a hybrid user model. In this hybrid user model, we use multi-attribute utility theory. We take advantage of the existing research on predicting query performance and on determining dissemination thresholds to create the functions to evaluate these chosen attributes. The main contribution of this research lies with the integration of user intent and system elements in a decision theoretic framework. Our approach also offers fine-grained representation of the model and the ability to learn a user's knowledge dynamically over time. We compare our approach with the best traditional approach in the information retrieval community - Ide dec-hi using term frequency inverted document frequency weighting on selected collections from the information retrieval community such as CRANFIELD, MEDLINE, and CACM.

Introduction

We study the problem of constructing a user model for improving a user's effectiveness in an information retrieval (IR) application. This problem has been investigated since the late 80s (Brajnik, Guida, & Tasso 1987; Saracevic, Spink, & Wu 1997) to address the lack of interests in users from the traditional IR framework by modelling a user's needs and retrieving more documents relevant to an individual user. The current approaches to building user models for IR are classified into three main groups (Saracevic, Spink, & Wu 1997): *system-centered*, *human-centered* and *connections* (the latter of which we will refer in this paper as *hybrid* approaches). The methods belonging to the system-centered group focus on using IR techniques such as relevance feedback and query expansion to create a user model (Spink & Losee 1996; Efthimis 1996; Borlund 2003; Ruthven & Lalmas 2003). The main idea of these approaches is that they iteratively improve a user's query by adding more terms, and/or updating weights for existing terms which are learned from relevant and non-relevant documents. This model is only good for the current query and is completely reset as the user changes from one query to the next. In summary, there is no history of a user's search behaviors except for the current query. The methods belonging to the human-centered group focus on using human computer interaction (HCI) approaches to create a user model. The main techniques include capturing the changes in cognitive states of users in the process of relevancy judgments (Jarter 1992), as well as pre-search and post-search interviews to create a user model. One key problem that arises with the approaches in this group is that they are concerned with a user's behaviors but have little to say about *why* a person might engage in one particular behavior. In order to find out *why*, we have to establish the relationships between the behaviors and the problems, and the relationship between a user's goals and a user's sub goals which are unfortunately missing from the group of human-centered approaches.

Lastly, methods belonging to the hybrid group combine the techniques from Artificial Intelligence (AI), HCI and IR to build a user model. Some research in IR and User Modeling (UM) do represent the hybrid view that bridges the system-centered and user-centered approaches (Logan, Reece, & Sparck 1994; Saracevic 1996) in an effort to resolve the weakness of both the system-centered and user-centered approaches. However, as Saracevic and his colleagues have succinctly pointed out (Saracevic, Spink, & Wu 1997), there is very little crossover between IR and AI/HCI communities with regards to building user models for IR. Since then, there have been several attempts recently from these communities trying to fill in this gap (for example: (Ford *et al.* 2002; Ruthven, Lalmas, & van Rijsbergen 2003; Billsus & Pazzani 2000; Smyth *et al.* 2004). However, the majority of the work on both sides is still focusing either on system-focused objectives only or user-focused objec-

*The work presented in this paper was supported in part by the National Geospatial Intelligence Agency Grant No. HM1582-04-1-2027 and UWW Grant for Undergraduate Research.

tives. This is quite unfortunate because many evaluation testbeds and methods are often re-invented by both sides. It is also very difficult to compare different techniques with each other because of the lack of unified evaluation procedures and metrics.

In this paper, we incorporate user-centered and system-centered approaches for building a hybrid user model for IR. We use well-established concepts and procedures in IR with the strength of knowledge representation techniques in AI to capture a user's intent in a search and combine the captured user intent with the IR system element in a decision theoretic framework. The goals of this model are: (i) To use the research in IR and research in UM in a decision theoretic framework to predict the effectiveness of the next retrieval task; and (ii) To allow a user to influence at a deeper level of an IR system rather than just at the query level.

This hybrid user model provides the missing information about the user to the system through user intent and provides the missing knowledge about the IR system to a user through a set of system elements. Even though there is some work from both the IR and UM communities which make use of decision theory (for example: (Balabanovic 1998; Brown 1998; Cooper & Maron 1978)), a decision theoretic framework that fully integrates attributes describing a user and attributes describing an IR system has not been explored before. We evaluate this hybrid user model and compare it with the best traditional approach in the IR community - Ide dec-hi using term frequency inverted document frequency weighting on selected collections from the IR community such as CRANFIELD, MEDLINE, and CACM. The results show that we retrieve more relevant documents in the initial run compared to the traditional approach.

This paper is organized as follows: We start with the related work section. Next, we provide the background of this model and describe our hybrid user model. We then discuss our evaluation of the external effect of our model on improving a hypothetical user's effectiveness in a search. We conclude by discussing ongoing and future extensions of this work.

Related work

Since we are focusing on developing a hybrid user model for an IR application, we now discuss some related work on hybrid methods in UM and IR. One of them is the work presented in (Logan, Reece, & Sparck 1994), in which Galliers theory of agents communications is applied in the MONSTRAT model (Belkin 1993). The Monstrat model specifies ten functions that an IR system needs to perform in order to achieve its goal of helping the user with his problem. Another work which partly inspired our effort is the STRATIFIED model proposed by Saracevic (Saracevic 1996) which resolves the weakness of both the system-centered and human-centered approaches. In the STRATIFIED model, both the user and the system sides are viewed as several levels of *strata*. Any level of the user's strata is allowed to interact with any level of the system's strata. This model is constructed based on the assumption that the interactions between the user and the target IR system do help the user's information seeking tasks.

In the recent years, researchers studying relevance feedback and query expansion have used a user's search behaviors for constructing a user model. The studies which incorporate a user's search behaviors into an IR process have shown that by understanding a user's search behaviors, we develop a more flexible IR system with personalized responses to an individual's needs (Campbell & van Rijsbergen 1996; Ruthven, Lalmas, & van Rijsbergen 2003; Spink, Greisdorf, & Bateman 1998). For example, the *ostensive model* in (Campbell & van Rijsbergen 1996) uses temporal factor and uncertainty associated with the assessment of individual document as evidence of relevance feedback process. The main idea of the ostensive model is that it treats the set of relevant documents as an *ordered set* with respect to the time when a user has assessed each document. The traditional probabilistic model would treat this set as *unordered set*. Therefore, the probability of a document being classified as relevant will be increased if this document has been assessed as relevant most recently.

The main difference between the existing approaches which incorporate the user's search behaviors with the approach presented in this paper is that they use the user's search behaviors to modify the *weight of an individual term* or *similarity measure* while ours uses the captured user intent to modify the *relationships among terms* of a query.

Background

User models are needed on top of an IR system because the traditional IR framework does not involve much input from a user except a user's query and some relevance feedback. Without a user model, it is very difficult to determine and update a user's needs. For instance, a user is searching for *"sorting algorithms"* and he possesses knowledge on *"distributed computing"* with an emphasis on *"parallel algorithms"*. He prefers to retrieve the papers on specific algorithms rather than the survey papers. He also prefers to retrieve as many potentially relevant documents as possible. For this user, a good IR system would display documents on parallel algorithms such as *Odd-Even transposition sort* or *shearsort* well *before* the sequential sorting algorithms such as *bubble sort* or *quick sort*. In other words, a good IR system would proactively modify the original request of "sorting algorithms" to a request on *parallel sorting algorithms* which connects the user's preferences, interests, and knowledge with his current request. Additionally, in order for him to see many potentially relevant documents, the threshold for filtering irrelevant documents should be set very low.

Our goal is to improve the effectiveness of a user engaged in an information seeking task by building a user model that integrates information about a user and an IR system in a decision theoretic framework. The components of a typical IR system include *query*, *indexing scheme*, *similarity measure*, *threshold*, and *collection*. Query represents a user's request. Indexing schemes contain domain knowledge represented in hierarchical relations of terms. Similarity measures are a function which determines how similar a user's query and a document from the searched collection is. Threshold is a real number which indicates how we should filter out irrelevant documents. A collection usually consists of a set of documents in a specific topic such as computer science or aerodynamics. Usually, these components are determined when the system is developed and used. Therefore, in order to build our hybrid model, our job now is to determine information about a user. We capture *user intent* in an information seeking task. We partition it into three formative components: the Interests capture *what* a user is doing, the Preferences captures *how* the user might do it, and the Context infers *why* the user is doing it. This section provides the description of the process of capturing user intent to build our *IPC model*. In the next section, we extend this model to create our hybrid user model. In this *IPC* model, we capture the Context, the Interests, and the Preferences aspects of a user's intent with a *context network* (*C*), an *interest set* (*I*), and a *preference network* (*P*). A context network *(C)* is a directed acyclic graph (DAG) that contains *concept nodes* and *relation nodes*. Concept nodes are noun phrases representing the concepts found in retrieved relevant documents (e.g *"computer science"*). Relation nodes represent the relations among these concepts. There are two relations captured: set-subset (*"isa"*) and relate-to relations (*"related to"*). We construct *C* dynamically by finding a set of sub-graphs in the intersection of all retrieved relevant documents. Each document is represented as a *document graph* (DG), which is also a DAG. We developed a program to automatically extract DG from text. Figure 1(a) shows an example of a context network for an analyst who is searching for information on terrorism and suspicious banking transactions. The Interests capture the focus and direction of the individual's attention. It is captured in the interest set (*I*). Each element of *I* consists of interest concept *(a)* and interest level *(L(a))*. An interest concept represents the concept that an analyst is currently focusing on while an interest level is any real number from 0 to 1 representing how much emphasis he places on this particular concept. *I* is initially determined from the current query, and the set of common sub-graph. Figure 1(b) shows the example of an interest set of the above analyst. Lastly, the Preferences describe the actions needed to perform to achieve the goals. We capture Preferences in a Bayesian network (Jensen 1996) which consists of three kinds of nodes: precondition (*Pr*), goals (*G*) and action nodes (*A*). Each node has two states: *true*, and *false*. Precondition nodes represent the requirements to achieve the goal nodes. Goal nodes represent the tools that are used to modify a users query. We currently have the two tools: filter which narrows down a query semantically and expander which broadens up a query semantically. The conditional probability table of each goal node is similar to the truth table of logical AND. Each *G* is associated with only one *A*. The probability of *A* is set to 1 if the tool is chosen and to 0, otherwise. Figure 1(c) shows an example of a preference network for the above analyst. The pre-condition nodes in this example consist of interest concepts such as *bank account*, *deposit*, and current query nodes. These nodes will be set as evidences (*true*) if they are belong to the current interest set or fully/partially matched with the current query. The filter or expander nodes simply mean that the action node associating with them will contain a link to a query graph that is narrower or broader than the original query graph.

P is updated when a user gives feedback. Basically, we add to *P* the tool that helps in the previous retrieval processes. If the total number of retrieved relevant documents exceeds a user-defined threshold, a tool is considered helpful.

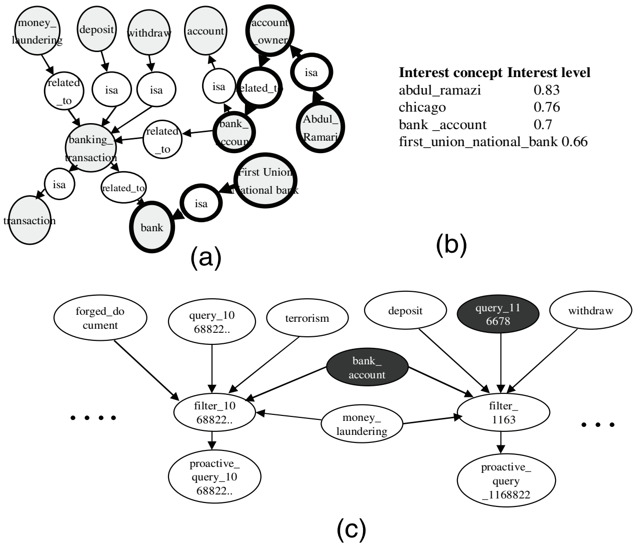
When a user issues a query q, it will be converted to a query graph (QG) which has the same representation as DG. The query graph is modified by using information from a user's Interests *I*, Preferences *P*, and Context *C* as follows:

- We set as evidence all interest concepts found in *P*. Find a pre-condition node *Pr* representing a query in *P* which has associated query graph (QG) that completely or partially matches against the given q. If such a node *Pr* is found, set it as evidence.
- Perform belief updating on *P*. Choose top n goal nodes from *P* with highest probability values (*SG*).
- For every goal node *g* in *SG*: If the query has been previously submitted and the user has used *g*, replace the original query sub-graph with the graph associated with the action node of this goal. If the query has not been asked before and *g* represents a filter: For every concept node q_i in the user's query graph *q*, we search for its corresponding node cq_i in *C*. For every concept a_i in *I*, we search for its corresponding node ca_i in *C* such that ca_i is an ancestor of cq_i. If such ci_i and cq_i are found, we add the paths from *C* between these two nodes to the modified query graph. It works similarly with an expander except that ca_i should be a progeny of cq_i.

The modified QG is sent to the search module where it is matched against each DG representing a record in our database. Those records that have the number of matches greater than a user-defined threshold are chosen and displayed to a user. A match between a QG q and a DG d_i is defined as $sim(q, di) = \frac{n}{2*N} + \frac{m}{2*M}$ in which *n, m* are the number of concepts and relation nodes of q found in d_i, respectively. *N,M* are the total number of concept and relation nodes of q. Two relation nodes are matched if and only if at least one of their parents and one of their children are matched. For more detail about our approach, please see our papers (Santos, Nguyen, & Brown 2001; Santos *et al.* 2003)

Hybrid User Model

We extend the *IPC* model by combining the user intent with the elements of an IR application in a decision theoretic framework to construct the hybrid model to improve a user's effectiveness in a search. By "a user's effectiveness", we refer to the effectiveness of an IR system with respect to the current searching goal. This can be determined quantitatively by using a function called *effectiveness function (F_e)*. An example of such a function can be *precision*, which is the ratio of the number of retrieved relevant document over the number of retrieved documents. Unfortunately, the computation of F_e is post-retrieval. It means that we only have enough information to compute the effectiveness of a search after the IR system has returned a set of documents to the user. In this hybrid user model, a pre-retrieval mechanism



Interest concept	Interest level
abdul_ramazi	0.83
chicago	0.76
bank _account	0.7
first_union_national_bank	0.66

Figure 1: (a) Context network (b) Interest set (c) Preference network

to estimate F_e is needed so that we can choose the solution that will likely improve F_e in the future retrievals.

Our solution is to convert this problem into a multi-attribute decision problem and use multi-attribute utility theory (Keeney & Raiffa 1976) in which a set of attributes is constructed by combining the set of attributes describing a user's intent and the set of attributes describing an IR system. In multi-attribute utility theory, the decision is made based on the evaluation of *outcomes* of the actions performed by a user or an agent. In this problem, the outcome space represents the set of *all* possible combinations of information about a user and information about a system and each outcome represents a specific combination of information about a user and information about an IR system. Let's take a look at the example in the background section. An outcome for that example may consist of a user's Interests to be *parallel algorithms*, his Context that contains relationships among *distributed computing, parallel algorithms*, and *sorting algorithms*, his query to be *sorting algorithms*, and his Preferences to be narrowing the original query and the threshold being low. There are two reasons for using multi-attribute utility theory. First, the estimation of the effectiveness function (F_e) with respect to the searching goal, in essence, is a problem of preference elicitation because it represents a user's preferences over a space of possible sets of values describing a user and describing an IR system. Second, the framework of a multi-attribute decision problem allows us to use the elicitation techniques from the decision theory community to decide which combination will likely produce the best effectiveness function.In this hybrid user model, eight attributes are initially considered from the set of attributes describing user intent and the set of attributes describing an IR system. They are: *I* (a user's Interest), *P* (a user's Preferences), *C* (a user's Context), *In* (Indexing scheme), S (similarity measure), T (threshold), D (Document collection) and *Q* (a user's query).

A straightforward approach is to list all possible outcomes available in the outcome space and then use a function to evaluate each outcome. This process is very tedious and time-consuming. In order to speed up the elicitation process, we need to reduce the number of attributes as much as possible. One way to achieve this goal is to find out the dependency among the attributes. Besides, we can always remove the attributes that are the same for all outcomes because they do not contribute to the decision making process. From the list of the above 8 attributes, we know that *I*, *P* and *C* have been captured in the *IPC* model (Santos *et al.* 2003) and have been used to modify a user's query *Q*. Therefore, the attribute *Q* subsumes the attributes *I*, *P* and *C*. In the traditional IR framework, the indexing scheme *In* is computed once when designing a system and shall remain unchanged during a search process. Therefore, *In* did not participate in the decision making process and should be removed. Similarly, documents are unchanged in the traditional IR framework and therefore *D* did not participate in the decision making process. Similarity is usually determined in the design phase of an IR system and thus is left out too. Even though there are some attributes that do not directly contribute to the decision making process, it is important that we assess the role for each attribute and justify our choices convincingly. After reducing the number of attributes to core attributes, we focus on only two attributes *Q* and *T*. We evaluate each outcome by a real value function. We make another assumption here that these two attributes are preferentially independent. Thus, this value function representing a user's preferences over these two attributes can be constructed as follows:

$$V(Q,T) = \lambda_1 V_1(Q) + \lambda_2 V_2(T)$$

where λ_i represents the importance of attribute *i* to the user, and V_i is a sub-value function for the attribute *i* with *i=1 or i=2*. This value function is generic for all IR systems and all type of users.

In this hybrid user model, we do not work directly with the value functions because it is very difficult to elicit the coefficients λ_i. Instead, we determine the partial value function which consists of two sub-value functions: one over query, and one over threshold.

The partial value function implies that an outcome x_1 with the value (x_{11}, x_{12}) is preferred to an outcome x_2 with value (x_{21}, x_{22}) if and only if

- $x_{1i} \geq x_{2i}$ for all *i=1,2*, and
- $x_{1i} > x_{2i}$ for some i.

For each sub-value function, each attribute is needed to be evaluated with respect to a user's effectiveness in achieving a searching goal at a given time. We assume that a user's searching goal at any given time is to retrieve many relevant documents quickly for the user. Therefore, we choose the average precision at fixed point recalls as the effectiveness function because it measures both the percentage of retrieved relevant documents and the speed of retrieving these documents.

Sub-Value Function over Query

We take advantages of the research on predicting query performance in the IR community to construct a sub-value function over a query. Basically, we have chosen the standard deviation of a query's terms' *inverted document frequency (idf)* as the core of this sub-value function. The main idea of *idf* measure is that the less frequent terms in a collection are the terms with more discriminating power. The main reasons for our choice are (i) the standard deviation of *idf* of a query's terms (also known as the distribution of informative amount in query terms (He & Ounis 2004)) has shown relatively good positive correlation with the average precision metric, and (ii) it can be computed in pre-retrieval process. We also verify this correlation with one of our experiments on the CRANFIELD collection (Cleverdon 1967). We found that the Spearman's correlation between the standard deviation of *idf* of a query's terms and average precision to be 0.323 (with average query length=9.06).

Recalling that each query in this approach is represented by a query graph (Santos *et al.* 2003) as described in the background section. Therefore, each query graph contains concept node and relation nodes. Therefore, we tried the sub-value functions for the concept nodes and for the relations. A sub-value function for the concept nodes is computed as follows:

$$V_c(Q) = \sigma_{idf-c}(Q) \qquad (1)$$

in which

$$\sigma_{idf-c}(Q) = \sqrt{\frac{1}{n}\sum_{c \in Q}(idf_c(c) - \mu_{idf_c}(Q))^2}$$

with *n* is the number of concepts in *Q*.

$$\mu_{idf-c}(Q) = \sum_{c \in Q}\frac{idf_c(c)}{n}$$

and

$$idf_c(c) = \frac{log_2(N+0.5)/N_c}{log_2(N+1)}$$

where *N* is the total number of documents in a collection and N_c is the total number of documents containing the concept *c*.

Similarly to the sub-value function computed based on information about concept nodes, we define sub-value function computed based on information about the relation nodes. A relation *r* in *Q* is represented as a tuple (c_1, r, c_2) in which c_1 and c_2 are two concept nodes, and *r* is either *"isa"* or *"related to"* relation.

$$V_r(Q) = \sigma_{idf-r}(Q) \qquad (2)$$

in which

$$\sigma_{idf-r}(Q) = \sqrt{\frac{1}{n}\sum_{r \in Q}(idf_r(r) - \mu_{idf-r}(Q))^2}$$

with *n* is the number of relation *r* in *Q*

$$\mu_{idf-r}(Q) = \sum_{r \in Q}\frac{idf_r(r)}{n}$$

and

$$idf_r(r) = \frac{log_2(N+0.5)/N_r)}{log_2(N+1)}$$

where *N* is the total number of documents in a collection and N_r is the total number of documents containing the relation *r*.

Sub-Value Function for Threshold

We take advantage of research from adaptive threshold in information filtering, specifically the work in (Boughanem & Tmar 2002) to construct a sub-value function for thresholds. We choose the threshold of the last document seen by a user and the percentage of returned documents preferred to be seen by a user as the core of our sub-value function.

For each query, the initial threshold can be determined as:

$$T_0 = p * N_0$$

where N_0 is the number of documents returned at time *0*, *p* is the percentage of retrieved documents that a user wants to see, for example, highest 10%, highest 20% or highest 80% of retrieved documents. For the first time when a user is using the system, this number is elicited by directly asking the user. If this is not the first time, then *p* is determined as follows:

$$p = \frac{l}{L}$$

where *l* is the number of documents that are returned in the previous retrieval and seen by the user and *L* is the number of documents that contain at least one concept in the query of the previous retrieval.

The threshold is updated by using one approach reported in (Boughanem & Tmar 2002):

$$T_{(t+1)} = T_t + \frac{sim(d_{last}) - T_t}{e^{\frac{(R_t-\lambda)}{\phi}}}$$

where $\lambda = 1300$ and $\phi = 500$ and R_t is the total number of relevant document at time *t*, d_{last} is the similarity of the last retrieved document in the previous retrieval. The values of these λ and ϕ constants are obtained from the experimental results in (Boughanem & Tmar 2002). The logic for this approach is that if the number of retrieved relevant documents is small, and the difference between the similarity of the last returned documents and the threshold is big, then we need to decrease the threshold considerably in order to retrieve more relevant documents. Otherwise, we can decrease the threshold a little.

This method of updating threshold is chosen because it is light-weight and can be computed in the pre-retrieval process. It also has been shown to correlate well with average precision in (Boughanem & Tmar 2002).

The sub-value function for the threshold attribute will then be defined as follows:

$$V(T) = \begin{cases} 1 & \text{if } T > T_t \\ 0 & otherwise \end{cases}$$

Complexity and Implementation of Hybrid User Model

The process of computing $idf_c(c)$ for every concept and every relation can be done offline. The complexity of this process is $O(nm)$ with *n* being the number of documents and *m* being the maximum number of nodes in a document graph. The only online algorithms are the computation of $V_c(Q)$ and $V_r(Q)$ for those concepts and relations included in a user's query. The computation of $V_c(Q)$ has complexity of $O(l_c log_2(N) + l_c)$ with l_c being the number of concepts in a query and *N* being the number of concepts in the collection. Similarly, the computation of $V_r(Q)$ has complexity of $O(l_r log_2(N) + l_r)$ with l_r being the number of relations in a query, and *N* being the number of relations in the collection.

Implementation (Work Flow)

The hybrid user model is integrated with the *IPC* user model as follows:

- A user logs into an IR system. If the user is new, then he/she is asked for his/her preferred percentage of documents needed to be returned *p*.
- The user issues a query *Q*. The user's query is modified using the information contained in the Interest, Preference and Context. Assuming that there are *m* goals fired in the Preference network, each goal generates a query, so we have the query sets $\{Q_1, Q_2, ..., Q_m\}$.
- Use the sub-value function to evaluate each Q_i. Choose the query with the highest sub-value function evaluation. Determine T_0 for initial threshold.
- Send the query with the highest value evaluated by the sub-value function to the search module, perform the search, filter our the documents based on the value of the threshold, and display the results to the user.
- After reviewing papers, we update the sub-value function *V(T)*. If a new query is issued, re-compute the threshold depending on the number of documents seen in the previous step.

Evaluation

The first objective of this evaluation is to assess whether the hybrid user model improves a hypothetical user's effectiveness in an information seeking task. Secondly, we would like to compare our hybrid user model with the existing approaches from the IR community by using collections, metrics and procedures from the IR community. In our previous papers (Nguyen *et al.* 2004b; 2004a), we have compared our *IPC model* against the Ide dec-hi with TFIDF (Salton & Buckley 1990) using the MEDLINE, CACM and a set of queries from the CRANFIELD collections. Therefore, in this evaluation, we re-use these collections as a testbed. We followed the standard procedure for evaluating any relevance feedback technique as described in (Salton & Buckley 1990). However, the standard procedure did not provide a way to assess the special features of our hybrid model. Thus, we employ a new evaluation procedure to assess the use of knowledge learned over time to modify queries and refer to it as *procedure to assess long-term effect*.

Testbeds

In this subsection, we describe in detail the testbeds used in this evaluation so that it is easy for readers to follow. MEDLINE, CACM and CRANFIELD are chosen as our testbeds in this evaluation. We chose these collections because they have been used widely in the IR community to evaluate the effectiveness of relevance feedback techniques (Salton & Buckley 1990; Loper-Pujalte, Guerrero-Bote, & Moya-Anegon 2003; Drucker, Shahrary, & Gibbon 2002).

In particular, CRANFIELD contains 1400 documents and 225 queries on aerodynamics; CACM contains 3204 documents and 64 queries in computer science and engineering (CSE); while MEDLINE contains 1033 documents and 30 queries in the medical domain (Salton & Buckley 1990). We use the complete set of queries from these collections in our evaluation.

Procedures

Standard procedure : We apply the standard procedure used in (Salton & Buckley 1990) for both Ide dec-hi/TFIDF and the IR application enhanced by our hybrid model. We issue each query in the testbed, we identify the relevant and irrelevant documents from the first 15 returned documents, and use them to modify the query proactively. For the Ide dec-hi/TFIDF, the weight of each word in the original query is re-computed using its weights in relevant documents and the first irrelevant document. The words with the highest weights from relevant documents are also added to the original query. For our user modeling approach, we start with an empty user model and add the concept and relation nodes to the original QG based on the procedure described in previous sections. We choose to use the sub-value function $V_c(Q) = \sigma_{idf-c}(Q)$ over concept nodes in a query as a sub-value function for the query because it is simple and easy to implement. In our preliminary evaluation of several value functions (Nguyen 2005), there is insufficient evidence to support the use of one sub-value function over the others. One good implication from this finding is that we can use a simple sub-value function, such as $V_c(Q)$, over concept nodes in a query and still achieve relatively good results. We then run each system again with the modified query. We call the first run, *initial run* and the second run, *feedback run*. For each query, we compute average precision at three point fixed recall (0.25, 0.5 and 0.75). We note that the CRANFIELD collection contains information about relevant and irrelevant documents while the other two collections contain only information about relevant documents.

Procedure to assess long-term effect : In this procedure, we would like to assess the effect of knowledge learned from a query or a group of queries. We start with an empty user model and follow the similar steps as described in the standard procedure above. However, we update the initial user model based on relevance feedback and we do not reset our user model, unlike the standard procedure above.

Results and Discussions

The average precision at three point fixed recall of the initial run and feedback run using original collection of the experiments in standard procedure for CRANFIELD, CACM and MEDLINE is reported in Table 1. Also in this table, we report the results for TFIDF/ Ide dec-hi approach. Note that in Table 1 and Table 2, "I" denotes *initial run* while "F" denotes *feedback run*. In the standard procedure, it shows that we achieve competitive performance using CACM collections compared to Ide dec-hi with TFIDF. For the CRANFIELD collection, we outperform TFIDF/Ide dec-hi approach in both runs. For the MEDLINE collection, we achieve clearly better results in the initial run compared to TFIDF approach.

	CRANFIELD		CACM		MEDLINE	
	I	F	I	F	I	F
Ide dec-hi	0.083	0.134	0.091	0.2	0.39	0.54
Hybrid	0.167	0.233	0.108	0.22	0.507	0.546

Table 1: Average precision at three point fixed recall for our hybrid user model with the standard procedure

The results of our procedure to assess long term effect of our hybrid approach are shown in Table 2. It shows that by using our hybrid model, the precision of the feedback runs is always higher than those of the initial runs. For the MEDLINE collection, for example, our initial run using knowledge of learned queries is even better than the feedback run of Ide dec-hi/TFIDF. That means the quality documents are retrieved earlier in the retrieval process than the other approach. For the CRANFIELD collection, we outperform the TFIDF/ Ide dec-hi approach in both initial and feedback runs. For the CACM collection, with the new procedure, we maintain the trend of retrieving more relevant documents in the initial run compared to TFIDF approach(0.144 vs 0.091).

	CRANFIELD		CACM		MEDLINE	
	I	F	I	F	I	F
Hybrid	0.175	0.237	0.144	0.256	0.587	0.67

Table 2: Average precision at three point fixed recall for our hybrid user model with the procedure to assess long term effect

In the past, we performed the procedure to assess long term effect using our *IPC model* over the entire CACM and MEDLINE collections (Nguyen *et al.* 2004a), as summarized in Table 3 below for easy comparisons. If we compare the results for *IPC model* (shown in Table 3) with the results for hybrid models (shown in Tables 1 and 2), we achieve only competitive results in both runs for the CACM collection while we are clearly better in the initial run and competitive for feedback run for the MEDLINE collection. Finally, we note that in previous results, we had to construct 27 query graphs out of 30 queries manually for the MEDLINE collection and 21 query graphs out of 64 queries manually for the CACM collection while in this version, we have improved our implementation for constructing document graphs from natural language text. Thus, every query graph has been automatically generated. [1]

	CACM		MEDLINE	
Proc.	I	F	I	F
Standard	0.095	0.223	0.4	0.583
Long-term	0.095	0.223	0.446	0.614

Table 3: Average precision at three point fixed recall for our IPC model (Nguyen *et al.* 2004a)

Future Work

In this paper, we have reported our approach of constructing a hybrid user model by combining both user-centered attributes and system-centered attributes in a decision theoretic framework. We present our first evaluation to assess its effectiveness in improving a hypothetical user's performance in an information seeking task using the CRANFIELD, CACM and MEDLINE collections. The results show that for the CRANFIELD collection, it outperformed the best traditional approach for relevance feedback in both runs; for the MEDLINE collection, it clearly achieved better results in both runs; and finally it achieves competitive results in the CACM collection in both runs. There are several issues that we are currently addressing for this research. First, we would like to combine the use of prior knowledge and knowledge of learned queries in an intuitive manner. Currently, we are conducting several experiments using different sets of *"seed"* user models which are created manually by users or semi-manually using system and users' preferences. The experiments will likely be finished in the next month. Secondly, in this evaluation, we only evaluate the sub-value function on the query but did not have a chance to assess how the sub-value function on threshold works. We would like to combine the evaluation with real users and assess the sub-value function on thresholds.

References

Balabanovic, M. 1998. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction* 8:71–102.

[1] As in the earlier experiments, document graphs were automatically generated but no effort was made at correcting them. The same approach continues in this current evaluation. As such, sometimes no document graphs were built but we do note that with our improved implementation for constructing document graphs, reduced the number of such occurrences.

Belkin, N. J. 1993. Interaction with text: Information retrieval as information seeking behavior. *Information Retrieval* 10:55–66.

Billsus, D., and Pazzani, M. 2000. User modeling for adaptive news access. *Journal of User Modeling and User-Adapted Interaction* 10:147–180.

Borlund, P. 2003. The concept of relevance in information retrieval. *Journal of the American Society for Information Science and Technology* 54:913–925.

Boughanem, M., and Tmar, M. 2002. Incremental adaptive filtering: Profile learning and threshold calibration. In *Proceedings of SAC 2002, Madrid, Spain*, 640–644.

Brajnik, G.; Guida, G.; and Tasso, C. 1987. User modeling in intelligent information retrieval. *Information Processing and Management* 23(4):305–320.

Brown, S. M. 1998. *Decision Theoretic Approach for Interface Agent Development*. Ph.D. Dissertation, Air Force Institute of Technology.

Campbell, I., and van Rijsbergen, C. 1996. Ostensive model of developing information needs. In *Proceedings of the Second International Conference on Conceptions of Library and Information Science: Integration in Perspective (CoLIS 2)*, 251–268.

Cleverdon, C. 1967. The cranfield test of index language devices. In *Reprinted in Reading in Information Retrieval Eds. 1998*, 47–59.

Cooper, W., and Maron, M. 1978. Foundations of probabilistic and utility-theoretic indexing. *Journal of the Association for Computing Machinery* 25(1):67–80.

Drucker, H.; Shahrary, B.; and Gibbon, C. 2002. Support vector machines: relevance feedback and information retrieval. *Information Processing and Management* 38(3):305–323.

Efthimis, E. N. 1996. Query expansion. *Williams, M., ed. Annual Review of Information Science and Technology* 31:121–187.

Ford, N.; Wilson, T.; Foster, A.; Ellis, D.; and Spink, A. 2002. Information seeking and mediated searching. part 4: Cognitive styles in information seeking. *Journal of the American Society for Information Science and Technology* 53:728–735.

He, B., and Ounis, I. 2004. Inferring query performance using pre-retrieval predictors. In *Information Systems, Special Issue for the String Processing and Information Retrieval: 11th International Conference*, 43–54.

Jarter, S. P. 1992. Psychological relevance and information science. *Journal of the American Society for Information Science* 43:602–615.

Jensen, F. V. 1996. *An Introduction to Bayesian Networks*. Univ. College London Press, London.

Keeney, R. L., and Raiffa, H. 1976. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons.

Logan, B.; Reece, S.; and Sparck, J. 1994. Modeling information retrieval agents with belief revision. In *Proceedings of the Seventeenth Annual ACM/SIGIR Conference on Research and Development in Information Retrieval*, 91–100.

Loper-Pujalte, C.; Guerrero-Bote, V.; and Moya-Anegon, F. D. 2003. Genetic algorithms in relevance feedback: a second test and new contributions. *Information Processing and Management* 39(5):669–697.

Nguyen, H.; Santos, E. J.; Zhao, Q.; and Lee, C. 2004a. Evaluation of effects on retrieval performance for an adaptive user model. In *AH 2004: Workshop Proceedings - Part I. Eindhoven, the Netherlands*, 193–202.

Nguyen, H.; Santos, E. J.; Zhao, Q.; and Wang, H. 2004b. Capturing user intent for information retrieval. In *Proceedings of the 48th Annual Meeting for the Human Factors and Ergonomics Society, New Orleans, LA*, 371–375.

Nguyen, H. 2005. *Capturing User Intent for Information Retrieval*. Ph.D. Dissertation, University of Connecticut.

Ruthven, I., and Lalmas, M. 2003. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review* 18.

Ruthven, I.; Lalmas, M.; and van Rijsbergen, K. 2003. Incorporating user search behavior into relevance feedback. *Journal of the American Society for Information Science and Technology* 54:529–549.

Salton, G., and Buckley, C. 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41(4):288–297.

Santos, E.; Nguyen, H.; Zhao, Q.; and Pukinskis, E. 2003. Empirical evaluation of adaptive user modeling in a medical information retrieval application. In *Proceedings of the ninth User Modeling Conference*, 292–296. Johnstown. Pennsylvania.

Santos, E.; Nguyen, H.; and Brown, S. M. 2001. Kavanah: Active user interface for information retrieval application. In *Proceedings of 2nd Asia-Pacific Conference on Intelligent Agent Technology*, 412–423. Japan.

Saracevic, T.; Spink, A.; and Wu, M. 1997. Users and intermediaries in information retrieval: What are they talking about? In *Proceedings of the 6th International Conference in User Modeling*, 43–54. Springer-Verlag Inc.

Saracevic, T. 1996. Relevance reconsidered. In *Proceedings of the Second Conference on Conceptions of Library and Information Science (CoLIS 2)*, 201–218. Copenhagen, Denmark.

Smyth, B.; Balfe, E.; Freyne, J.; Briggs, P.; Coyle, M.; and Boydell, O. 2004. Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction* 14(5):383–423.

Spink, A., and Losee, R. M. 1996. Feedback in information retrieval. *Williams, M., ed., Annual Review of Information Science and Technology* 31:33–78.

Spink, A.; Greisdorf, H.; and Bateman, J. 1998. From highly relevant to not relevant: Examining different regions of relevance. *Information Processing and Management* 34:599–621.