# Modeling Users for Adaptive Information Retrieval by Capturing User Intent

Eugene Santos, Jr.
Thayer School of Engineering,
Dartmouth College,
8000 Cummings Hall
Hanover, NH 03755
E-mail: Eugene.Santos.Jr@dartmouth.edu

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

**Abstract:** In this chapter, we study and present our results on the problem of employing a cognitive user model for Information Retrieval (IR) in which a user's intent is captured and used for improving his/her effectiveness in an information seeking task. The user intent is captured by analyzing the commonality of the retrieved relevant documents. The effectiveness of our user model is evaluated with regards to retrieval performance using an evaluation methodology which allows us to compare with the existing approaches from the information retrieval community while assessing the new features offered by our user model. We compare our approach with the Ide dec-hi approach using term frequency inverted document frequency weighting which is considered to be the best traditional approach to relevance feedback. We use CRANFIELD, CACM and MEDLINE collections which are very popular collections from the information retrieval community to evaluate relevance feedback techniques. The results show that our approach performs better in the initial runs and works competitively with Ide dec-hi in the feedback runs. Additionally, we evaluate the effects of our user modeling approach with human analysts. The results show that our approach retrieves more relevant documents to a specific analyst compared to keyword-based information retrieval application called Verity Query Language.

**Keywords:** information retrieval, user model, user intent, relevance feedback, empirical evaluation, context, Bayesian networks, intelligence analysts.

## Introduction

We studied the problem of employing a user model for Information Retrieval (IR) in which knowledge about a user is captured and used for improving a user's performance. A user model addresses the "one size fits all" problem of the traditional IR system (Brusilovsky & Tasso, 2004). It takes into consideration a user's knowledge, preferences, interests, and goals of using an IR system to deliver corresponding documents that are relevant to an individual and to present different parts of the same documents to a user according to his/her preferred ways of perceiving information. Modeling a user in an information seeking task also addresses the gap between what a user thinks as relevant versus what an IR system assumes that any user would think as relevant (Saracevic et al., 1997). The main purpose of user modeling for IR is to determine what the user intends to do within a system's environment for the purpose of assisting the user to work more effectively and efficiently (Brown, 1998). The common approach for an IR application that employs a user model usually consists of two main steps: (i) to construct a static, or a dynamic user profile; and (ii) to adapt the target IR application to the user's profile. An example of a static user profile is his/her demographic data such as gender, age, profession, and zip code. An example of a

dynamic user profile is his domain knowledge, goals, and preferences. The first step is referred to as elicitation and the second step is referred to as adaptation. Elicitation of user models is a knowledge acquisition process. It is well-known in the artificial intelligence (AI) community that knowledge acquisition is the bottleneck of intelligent system design (Murray, 1997). Determining when and how to elicit the user's knowledge is a domain and application-dependent decision. Adaptation involves how to retrieve documents that are relevant to the user's profile and how to present these relevant documents according to the user's preferred ways of perceiving information.

User modeling techniques have been used to improve a user's performance in information seeking since the late 80s (examples of some early works are (Allen, 1990; Brajnik et al., 1987; Saracevic et al., 1997)). Modeling a user for information seeking poses many challenges to both the information retrieval and the user modeling communities. We have identified five main challenges as follows:

(i) the partial-observability of a user's knowledge (e.g. as identified in (Wilson, 1981)). A user's information needs is a subjective experience that only exists in a user's mind and therefore, it is not directly accessible to outsiders (Case, 2002;Wilson, 1981).

(ii) the uncertainty when modeling a user (e.g. as identified in (Greenberg & Witten, 1985; Chin, 1989)). Even within a very small domain, the number of possible actions that a user can perform may increase exponentially over time. To make matters worse, modeling every possible action in the user's world unfortunately does not lead to the most accurate model (DeWitt, 1995).

(iii) the vagueness of an individual's information needs (e.g. as identified in (Case, 2002; Wilson, 1981;)). These challenges are caused by a user's inexperience in problem solving, a user's unfamiliarity with the search subjects, or a user's lack of required computer skills. If a user does not know exactly what he/she is looking for, he/she often constructs queries with terms that are either too broad or too specific and are not closely related to what he/she actually needs.

(iv) the dynamics of a user's knowledge which changes over time as a result of new information (e.g. as identified in (Belkin, 1993; Ingwersen, 1992)). The traditional IR framework assumes that a user's information needs are static. This means that the content of retrieved documents did not have any effect on a user. However, studies have shown that a user's knowledge is updated over time by interacting with information (Ingwersen, 1992; Campbell, 1999).

(v) the absence of a standard, unified evaluation framework on the effectiveness of such a model (e.g. as identified in (Weibelzahl, 2003)). After all, the goal of a user model is to make an IR system better serve a user. Unfortunately, empirical evaluation is often overlooked even within the user modeling (UM) community (Chin, 2001). There are two main schools of thought regarding the evaluation of effectiveness of a user model from the UM and IR communities. The techniques from the UM community focus more on the issues of assessing an individual user's performance (Chin, 2003) whereas the techniques from the IR community focus more on the issues of assessing an IR system's performance. In the IR community, standard metrics such as precision and recall (Salton & McGill, 1983) and benchmark data collections are used to evaluate how many documents assessed

as relevant by a group of experts have been retrieved. In the last few years, the evaluation methods in the IR community have shifted towards more concerns for the end users, for example, the interactive track at TREC conference (Craswell et al., 2003; Wilkinson & Wu, 2004), but this is still in its infancy with limited attention and participation from the IR community. The reasons for this situation are that the evaluations involving real users are expensive, time-consuming, and contain a lot of measurement noise. The key to solving this problem is to develop testbeds, metrics and procedures that take into account a user's experience in a search as well as an IR system's performance.

Unfortunately, traditional IR does not offer a way to overcome these challenges because its framework supports very little users' involvement (Saracevic, 1996).

In Collaborative IR (Karamuftuoglu, 1998) and Social IR (Goh & Foo, 2007), modeling a user is a very critical issue. In order to do a good job in CIR and SIR, we need to understand a user's behaviors in an information seeking task. In other words, we need to capture an individual's experience, knowledge, interests, and cognitive styles and use this information to assist the user in stand-alone and collaborative retrieval tasks. The research on using user models for information retrieval will help answer some of the most important research questions for CIR and SIR: how to capture a user's behaviors in an information seeking task and how to improve a user's performance and satisfaction using the captured information.

The current approaches to building user models for information retrieval are classified into three main groups (Saracevic et al., 1997): system-centered, human-centered, and connections (which we will refer in this chapter 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. The methods belonging to the human-centered group focus on using human computer interaction (HCI) approaches to create a user model. The methods belonging to the hybrid group combine IR, HCI or Artificial Intelligence (AI) techniques to construct a user model. As Saracevic and his colleagues (1997) have succinctly pointed out, there is very little crossover between IR and AI communities with regards to building user models for IR. This is quite unfortunate because many techniques and evaluation methods are often reinvented by both sides. The main objective of our approach is to take advantage of well-established evaluation frameworks in IR, and use the strength of knowledge representation techniques in AI to build and evaluate a user model for improving retrieval performance.

In this chapter, we present our effort in improving a user's effectiveness in an IR application by building a user model to capture user intent dynamically. We assess its effectiveness with an evaluation methodology which allows us to compare with the existing approaches from the IR community as well as validates its effectiveness with human subjects. This chapter brings together some of our past results and user modeling experiments providing a unified formal framework and evaluations with synthesized data sets and human testing (Santos et al., 2003a; Santos et al., 2003b; Nguyen et al., 2004a; Nguyen et al., 2004b; Nguyen, 2005). Uncertainty is one of the key challenges in modeling a user for IR, as mentioned earlier. While there are some other approaches to modeling uncertainty such as Dempster-Shafer theory (Shafer, 1976), we selected Bayesian networks (Pearl, 1988) since it provides a mathematically sound model of uncertainty and we have expertise in efficiently building and reasoning over them (Santos et al., 2003c; Santos & Dinh, 2008). The novelty of our approach lies with the fine-grained representation of a user model, the ability to learn user knowledge incrementally and dynamically, and the evaluation framework to assess the effectiveness of a user model.

The goal of our user model is to capture a user's information seeking intent. We observe that a

document's content reflects its authors' intent (termed as author intent) because the authors want to convey certain messages through their writings. The intent of a user engaged in an information seeking task is reflected in his/her information needs. When a user indicates which documents are relevant to his/her needs, the user intent and the author intent are probably overlapped. Therefore, capturing a user's intent by analyzing the contents of retrieved relevant documents is an intuitive task to do. Moreover, a user's intent is characterized by the goals that the user is trying to achieve, the methods used to achieve them, and the reasons why a user wants to accomplish a certain goal. Therefore, we partition a user's intent into three formative components: interests, preferences, and context which captures what goals a user focuses on, how a user is going to achieve them, and why a user wants to achieve these goals. We refer to our user model as the *IPC* user model. While previous efforts at building a user model for information retrieval and filtering have either focused exclusively on learning any one of these aspects alone (e.g., Balabanovic & Shoham, 1997; Billsus & Pazzani, 2000; Horvitz et al., 1998; Hwang, 1999; Maes, 1994), or combining dynamic interests and/or local context with static ontology (e.g., Hernandez et al., 2007; Liu & Chu, 2007; Mylonas et al., 2008), we focus on generating an individual's context dynamically from the concepts relevant to his/her information seeking tasks. We construct our user model over time and incrementally from retrieved relevant documents as indicated by the users in an interactive IR framework. Our user model also focuses on interactions among a user's interests, preferences and context in a dynamic fashion. In particular, our emphasis is on deriving and learning the context for each user which is essentially the relations between concepts in a domain dynamically. The difference of our approach versus the existing work in this direction is that we provide a learning capability for the system to discover new knowledge based on analyzing the documents relevant to the user and their context, i.e., why a user is focusing on the given information by exploring the structure of information instead of frequency.

In our evaluation framework, we assess the effectiveness of our user model with regards to the target application in terms of its influence on retrieval performance as well as its effects on helping humans to retrieve more documents that are relevant to an individual's needs (Santos et al., 1999; Santos et al., 2003a; Santos et al., 2003b; Nguyen et al., 2004a; Nguyen et al., 2004b; Nguyen, 2005). We discuss the results of our evaluation on the effectiveness of our user model with regards to retrieval performance using the CRANFIELD, CACM, and MEDLINE collections (Salton & Buckley, 1990). We compare against the best traditional approach to relevance feedback – the Ide dec-hi approach using term frequency inverted document frequency (TFIDF) weighting (Salton & Buckley, 1990). Even though the Ide dec-hi approach is old, it is still very competitive, especially in evaluations with small data sets such as CRANFIELD, CACM and MEDLINE, as shown in more recent studies (Drucker et al., 2002; Lóper-Pujalte et al., 2003). The difference between our user modeling approach and the Ide dec-hi approach is that our model is long-lived and changes overtime while the user model created by using the Ide dec-hi is short-lived and only affects the current query. Therefore, we simulate the traditional procedure as laid out in (Salton & Buckley, 1990) by starting with an empty user model. We also create a new procedure in which we assess new features offered by our user model such as the use of prior knowledge and the use of information learned from one query to the same query and related queries. We show that our approach performs better than TFIDF in the initial run; works competitively with Ide dec-hi in the feedback run; and, improves retrieval performance when knowledge about users and search domains have been learned over time. In addition to the evaluation with synthesized data, we conduct an evaluation with human analysts. Our evaluation with three intelligence analysts was conducted at the National Institute of Standard Technology (NIST). The results show that our

approach retrieves more relevant documents that are unique for each user.

This chapter is organized as follows: We begin by reviewing important related work in IR and UM communities with regards to capturing user intent, using relevance feedback and Bayesian networks to improve retrieval performance. Next, our approach will be presented, followed by a description of the evaluation procedures with synthesized data and the analysis of the results. Then, a description of the evaluation with intelligent analysts will be reported. Finally, we present our conclusions and future work.

## Related work

Our approach builds a user model by capturing user intent for improving the user's effectiveness in an IR application. Our technique makes use of relevance feedback and Bayesian networks for building our user model dynamically and uses information about a user's interests, context and preferences to modify a user's query proactively. In this section, we first review some work on capturing user intent from the IR community and then present some related research on relevance feedback, Bayesian networks and context for IR.

According to the Webster dictionary (online version available at http://www.merriam-webster.com/, based on *Merriam-Webster's Collegiate® Dictionary, Eleventh Edition)*, intent is defined as "a usually clearly formulated or planned intention". We define intent as composed of the user's desired end-states (goals), the reason for pursuing such end-states, methods to achieve these goals, and the levels of commitment behind them. An important aspect behind intent is to capture as much as possible of the user's knowledge, experience, and individual cognitive style to improve a user's effectiveness in an information seeking task. In the IR community, researchers have recently focused on identifying the goals of a user's search to retrieve more relevant information to the tasks that the user is doing (Broder, 2002; Baeza-Yates et al., 2006; Jansen et al., 2007; Lee et al., 2005; Rose & Levinson, 2004). However, two questions need to be addressed in order to capture a user's intent:

(i) Whether a user has predictable goals in information seeking and which factors can affect the user's goals?

(ii) How can we identify a user's goals automatically?

In (Rose & Levinson, 2004), data collected from a user study has been analyzed and the goals of a user's queries are classified into two categories: navigational and informational. Navigational queries refer to the ones issued by a user who knows exactly which web pages he/she is going to visit. Informational queries refer to the ones issued by a user who does not have any specific web pages in mind and use IR applications for learning or exploring a specific topic. One more category added by Broder (2002) and later explored by Jansen et al. (2007) is transactional, which refers to "the intent to perform some web-mediated activity" (Broder, 2002). Another similar classification scheme is introduced in (Baeza-Yates et al., 2006), which include informational, not informational, and ambiguous. These categories can be determined manually (for example in (Broder, 2002; Jansen et al., 2007)) and automatically (for example, in (Lee et al., 2005), goals are determined using frequency information from click and anchor link distribution). Yahoo used to maintain a research site (http://mindset.research.yahoo.com) that allowed users to sort the retrieved web sites into commercial and non-commercial based on whether a user is shopping or seeking information. In our approach, we infer a user's intent automatically and dynamically using information from a user's queries and documents that they have indicated as relevant. Even though query is important, it may not be enough to represent a user's information needs because of various reasons, including vagueness due to the user's inexperience and partial-observability due to the

user's inability to map his/her information needs into words. Therefore, we believe that additional information such as document contents may shed some lights on the actual intention of a user.

Relevance feedback is an effective method for iteratively improving a user's query by learning from the relevant and non-relevant documents of a search (Spink & Losee, 1996). Relevance feedback and query expansion techniques have been used widely in the IR community since the early 60s (Frake & Baeza-Yates, 1992). Several comprehensive reviews of research in relevance feedback and query expansion are (Borlund, 2003; Efthimis, 1996; Ruthven & Lalmas, 2003; Spink & Losee, 1996). Not only did the IR community focus on the development of different techniques for improving retrieval performance using relevance feedback and query expansion, IR researchers also focused on the evaluation of the effectiveness of these two approaches early on. In particular, Salton and Buckley (1990) have laid out an evaluation framework to assess and compare any technique using relevance feedback and query expansion. Twelve different relevance feedback techniques including Ide dec-hi, Ide regular (Ide, 1971), and Rochio (1971) have been evaluated for vector space and probabilistic models using 6 collections: CACM, CISI, CRANFIELD, INSPEC, MEDLINE and NPL. Two important issues raised by this evaluation are the use of a residual collection and the computation of average precision at three specific recall points of 0.25, 0.5, and 0.75 (or as it is called, three point fixed recall). A residual collection is created by removing all documents previously seen by a user regardless of whether they are relevant or not from the original collection. Evaluation is done using the reduced collection only. The main idea for using the residual collection is to assure that we assess a technique based on new information retrieved (Salton & Buckley, 1990). The average precision at particular fixed recall points offers an opportunity for easy comparison among different techniques. The best technique found in this evaluation over all tested collections is Ide dec-hi in which the terms from all relevant documents and the first non-relevant document are added to the original query. Ide dec-hi is still very competitive in some newer studies, as pointed out in (Drucker et al., 2002; López-Pujalte et al., 2003). Relevance feedback or query expansion techniques represent a user's information needs explicitly by words or terms that the user is looking for or has found in relevant or non-relevant documents. This type of model captures only the user's immediate interests for a specific query. This model is only good for the current query and is completely reset as the user change from one query to the next. The content of the model is not reused if a similar query is issued. In summary, there is no history of user search behaviors except the relevance feedback for the current query.

In our approach, we made use of Bayesian networks to build our user model. Bayesian networks also have been applied to IR for improving ranking (deCristo et al., 2003) and improving relevance feedback and query expansion (Decampos et al., 1998; Haines & Croft, 1993). The techniques proposed in (deCristo et al., 2003; Haines & Croft, 1993) take advantage of the expressiveness of Bayesian networks to represent the relationships among queries, documents, and keywords while the approach in (Decampos et al., 1998) uses the inference power of Bayesian networks with relevance feedback set as evidence. In our approach, we focus on exploring the structure of information based on syntactic analyses of sentences and noun phrases instead of frequency as have been done in other Bayesian networks approaches above. We also use Bayesian networks to reason about a user's preferences in choosing a class of tools to modify a query.

The techniques for using feedback from users as inputs for machine learning mechanisms to build user models are widely used in information filtering and text recommendation systems. Please see Zukerman & Albrecht (01) for a comprehensive survey on statistical approaches. Some techniques have successfully captured a user's interests in information seeking such as (Balabanovic, 1998; Billsus & Pazzani, 2000). The work of Adaptive News Filtering (Billsus &

Pazzani, 2000) is a typical example of how to use machine learning techniques to build user models while the work by Balabanovic (1998) on a text recommender system is a typical example of using decision theoretic principles to represent preference rankings over a set of documents. Each of the two UM approaches above is evaluated using their own collections and evaluation procedures. Therefore, it is very difficult to compare them against different techniques. Other studies from UM community (Bueno & David, 2001; Magnini & Strapparave, 2001) also use their own users, collections and procedures.

Even though context in information retrieval is considered a very important research topic in the IR community (e.g., early work from Saracevic (1996) and Dervin (1997)), until recently, IR researchers have emphasized more on capturing contextual information for improving retrieval effectiveness (e.g., special issues on context for information retrieval edited by Cool and Spink (2002); another special issue edited by Crestani & Ruthven (2007); or, paper by Mylonas and colleagues (2008)). The studies presented in the special issue on context for information retrieval edited by Cool and Spink (2002) sheds some light on how to capture a user's context at different levels (e.g., environment, information seeking, interaction, and query levels). The studies in the special issue edited by Crestani and Ruthven (2007) are directly related to our work in this chapter. In particular, in the approach proposed by Campbell et al. (2007), a naive Bayesian classifier is created and used to find relationships between documents. One problem with this approach is that it requires many instances of training data for the classifier to start working. Another direction to explore contextual information is to use domain ontology to support text retrieval (Hernandez et al., 2007; Liu & Chu, 2007). Our approach is different from these two approaches in that we construct individual contextual information dynamically instead of using static domain ontology. Another work that is very closely related to our approach is Mylonas and colleagues (2008). In this work, domain ontology is combined with individual context information extracted from annotations, and queries are used to provide context information for IR. In our work, we currently extract individual context information from analyzing the whole content. However, our approach does work with annotations or snippets as well.

In summary, our approach is different from other approaches reviewed in this section in two aspects. First, we define and capture the user's intention in an information seeking task dynamically by exploring the structure information from retrieved relevant documents. Second, we focus on assessing whether our technique retrieves more relevant documents for individual user as well as comparing our results with the existing techniques from the IR community.

## IPC User Model

The main goal of our user model is to accurately capture and represent a user's intent in order for the main IR application to be able to assist the user in getting more relevant documents for the tasks at hand (Santos et al., 2001). We partition a user's intent in information seeking into three formative components. The first, Interests, captures the focus and direction of the individual's attention. The second, Preferences, describes the actions and activities that can be used to carry out the goals that currently hold the individual's attention, with a focus on how the individual tends to carry them out. The third, Context, provides insight into the user's knowledge and deeper motivations behind the goals upon which the individual is focused and illuminates connections between goals. In other words, Interests component captures what a user is doing, Preferences component captures how the user might do it, and Context component infers why the user is doing it. Within the context of an IR application, our user model uses the information captured on what a user is currently interested in, how a query needs to be constructed, and why the user dwells on a

search topic in order to modify a user's queries pro-actively and autonomously, and send the modified queries to our search engine to retrieve documents for the user.

In our user model, we capture the context, interest, and preference aspects of a user's intent with a context network, an interest set, and a preference network, correspondingly. Before describing the details of the IPC model and how queries are modified on behalf of the user, we first introduce the document and query representation called document graph and query graph. Each document is represented as a document graph (DG). A *DG* is a directed acyclic graph (DAG) in which each node represents a concept (called concept node) or a relation among the concepts (called relation node). Concept nodes are noun phrases such as "*surface heat transfer*" or "*transfer rate distribution*". Relation nodes are either nodes labeled "*isa*" or "*related to*". A relation node of a DG should have concept nodes as its parent and its child. The algorithm for extracting concepts and relations between these concepts from a text file is included in Appendix. A query graph (QG) is similar to the document graph but it is generated from a user's query. DG and QG will be used in our description of the algorithms for constructing our user model.

**Interest set**

The Interest set determines what is currently relevant to a user. Each element in the interest set consists of an interest concept and interest level. Interest concepts refer to the concepts the user is currently focusing on, and interest levels are real numbers from 0 to 1 representing how much emphasis the user has on a particular concept. The concept with an interest level of 1 is a concept of maximum interest while one with an interest level of 0 is of minimum interest to the user. An interest set is created and updated based on the intersection of retrieved relevant documents. Since a user's interests change overtime, we incorporate a fading function to make the irrelevant interests fade away.

The concepts in an interest set are determined from the set of documents that the user has indicated as relevant. Denote each interest concept as *a* and its associated interest level as *L(a)*. We compute *L(a)* after every query by:

$$L(a) = 0.5 * (L(a) + \frac{p}{q})$$

where *p* is the number of retrieved relevant documents containing *a* and *q* is the number of retrieved documents containing *a*. If *L(a)* falls below a user-defined threshold value, the corresponding interest concept *a* is removed from the interest set. After we find the set of DGs in the intersection of retrieved relevant documents, as described in the next section, each concept in this set will be added to the current interest set with the interest level being computed as the ratio of frequency of a specific concept over the total of concepts in the intersection.

**Context network**

The Context network captures a user's knowledge of concepts and the relations among concepts in a specific domain. Figure 1 shows a portion of the context network of a user model from one of the experiments conducted in this chapter. The representation of a context network is similar with that of a document graph. It is basically a directed acyclic graph (DAG) that contains two kinds of nodes: concept nodes and relation nodes. We use DAG to represent a context network because of its expressiveness in representing the relations among concepts in a specific domain visually. Each node is associated with a weight, value and bias. The weight of a node represents its importance assigned initially by the system. The concept nodes and "*isa*" relation nodes have initial weights equal to 1 while the "*related to*" relation nodes have initial weight equals to 0.8. We choose these

values to emphasize that with "*isa*" relation, one node can transfer all of its energy to the other node while with "*related to*" relation, only a part of its energy can be transferred to another node. The value of a node represents its importance to a user and is any real number from 0 to 1. The bias of a node represents whether this node is actually in the user's interests or not. Each node's weight, value and bias will be used by a spreading activation propagation algorithm to reason about concepts of a user's interests used in determining the modified query graph. The main idea is that a node that is located far from an evidently been interested concept will be of less interest to the user.

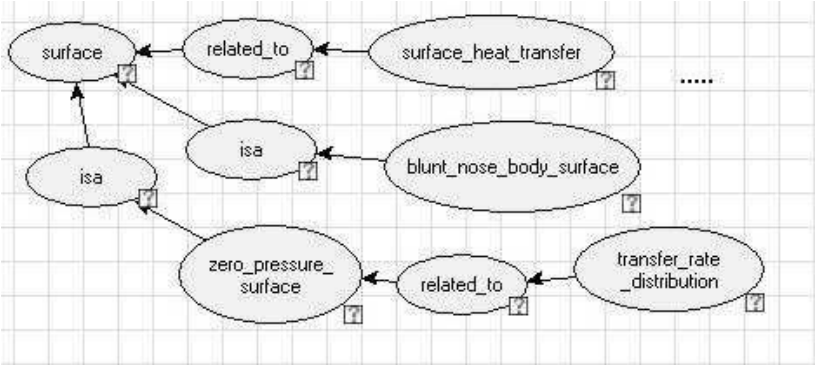



**Figure 1.** An example of a context network.

The spreading activation propagation algorithm consists of the followings:

- The inputs to this spreading activation algorithm are the current interest set, current query graph and current context network of our user model. The output of this algorithm is a new interest set.
- We set the bias equal to 1 for every concept found both in the current context network and in the current query graph. We set the bias equal to the interest level for every interest concept found both in the current context network and in the current interest set.
- We sort all the nodes based on its depth in the context network. Denote a node as $a$, its depth as $d(a)$.

  $d(a) = 0$ if the node has no parents

  $d(a) = \max(d(p(a))) + 1$ with $p(a)$ is a parent of $a$.
- For each node $a$ in the existing context network

  If this node doesn't have any parents:

  $sum = 0$

  $$value = \frac{(sum + bias)}{2} \qquad if \quad bias > sum$$

  $value = 0$ $\qquad\qquad$ *Otherwise*

  If this node has one parent node:

  $sum = value(p(a)) * weight(p(a))$

$$value = \frac{(sum + bias)}{2} \quad if \quad bias > sum$$

$$value = sum \qquad Otherwise$$

If this node has multiple parent nodes:

$$sum = \frac{1}{1 + e^{-\frac{\sum value(p_i(a)) * weight(p_i(a))}{n}}}$$

in which $p_i(a)$ is a parent of a node $a$, $n$ is the total number of all parent nodes. We chose this function to ensure that the value of each node is converged to 1 as the values and weights of its parents are increasing.

$$value = \frac{(sum + bias)}{2} \quad if \quad bias > sum$$

$$value = sum \qquad Otherwise$$

- Sort all concept nodes by their values and pick the nodes whose values are greater than a user-defined threshold to form a new interest list.

We treat nodes with one parent and multiple parents differently in this algorithm to stress the important influences a child node received from the only parent versus received from many different parents.

In the example shown in Figure 2, a user's query is about "*transfer rate distribution*". The node corresponding to the query term found in the existing context network has initial weight and bias being 1 (shown as a shaded node in Figure 2). We assume further that the existing interest list contains two concepts: "*surface*" with interest level being 0.6 and "*surface heat transfer*" with interest level being 0.7. We apply this algorithm and recompute the values for the concepts nodes in this context network. The value of the node "*surface*" is slightly increased from 0.6 to 0.6047 while the node "*surface heat transfer*" is sharply decreased from 0.7 to 0.35. The node "*surface heat transfer*" is located farther from the concept in the query node compared to the node "*surface*". If the threshold for values of interest list is set as 0.4 for example, then the new interest list will consist of only two nodes: "*transfer rate distribution*" and "*surface*".

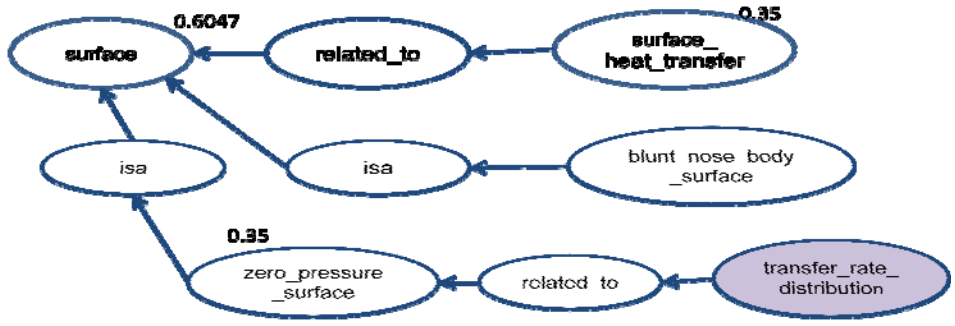



**Figure 2:** Example illustrates spreading activation algorithm on a context network

We construct a context network dynamically by finding the intersection of all document graphs representing retrieved relevant documents. The algorithm for finding intersections of retrieved relevant document graphs consists of the following:

Denote the set of retrieved relevant documents as $D=\{D_1,D_2,..D_m\}$.
Denote intersection set as $J$ which is initially empty.

For each document $D_i$ belongs to $D$ do
  For each node $c$ in $D_i$ do
    sum=0
    if this node is a concept node then
    For $j$ from $0$ to $m$
                if ($i$ does not equal to $j$) and ($Dj$ contains concept node $c$)
                then sum++
    if (sum ≥ user-defined-threshold)
                 then Add concept node $c$ to $J$
For each document $D_i$ belongs to $D$ do
  For each node $r$ in $D_i$ do
          sum =0
          if this node is a relation node and (its parent and its child are in $J$)
          then sum++
          if sum > 0 then add this fragment $r$'s parent - $r$ -$r$'s child to $J$

The common set of sub-graphs of the two documents described in Figure 3 is shown in Figure 4. The set of common sub-graphs is used to update our context network. We will check if a sub-graph is not currently in the context network, and add it accordingly. We will also ensure that the update will not result in a loop in the existing context network. If it does, we skip this addition. We avoid loops in our context network to make sure that our context network is a directed acyclic graph. A new link between two existing concepts in a context network will also be created if two concepts are indirectly linked in the set of common sub-graphs and the frequency of these links exceeds a certain user-defined threshold.

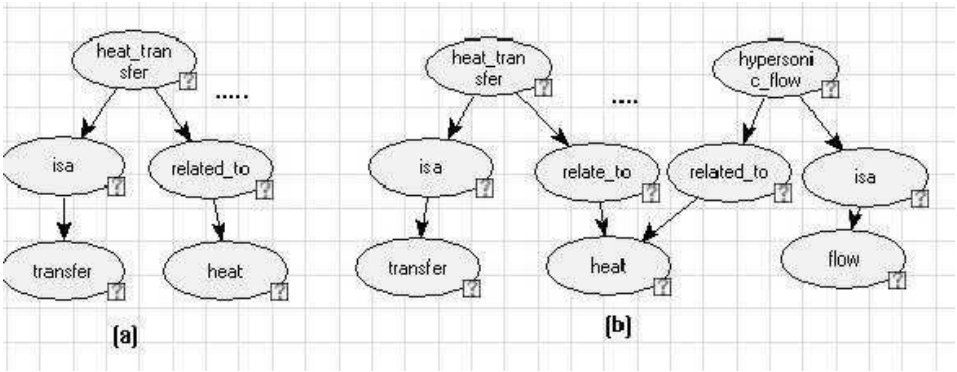



**Figure 3.** A portion of document graphs for two documents

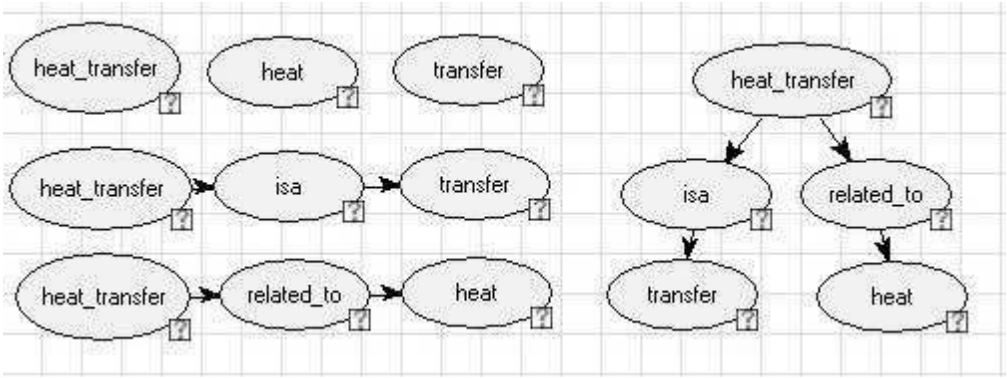



**Figure 4.** A portion of common set of sub-graphs of two documents in Fig. 3

**Preference network**

The Preference network represents how a user wants to form the query. We use Bayesian networks to represent a preference network because of its expressiveness and power for modeling uncertainty. A user's preference is reflected by how a user prefers to use a class of tools. A tool is defined as an operator to perform specific actions to transform the information that the user needs based on his preferences. There are three kinds of nodes in a preference network. The first type is a precondition node (*Pc*) which represents the requirements of a tool. A user's query and the concepts contained in the current interest relevancy set are examples of pre-condition nodes. The second type is a goal node (*G*) which represents a tool. An example of a tool is a filter that searches for documents that narrows down the search topics semantically. Another example of a tool is an expander that searches for documents that broaden up the search topics semantically. The third type of node is an action node (A) which is associated with each goal node. For each pre-condition node representing a user's current interest, its prior probability will be set as the interest level of the corresponding interest concept. The conditional probability table of each goal node is similar to the truth table for logical AND. In the current model, each goal node is associated with only one action node. The conditional probability of the action node will be set to 1 if the corresponding tool is chosen and to 0, otherwise.

A preference network is built when a user issues a new query and is updated when a user gives relevance feedback after each query. Every query is considered a pre-condition node in a preference network. If this query or some parts of it have been encountered before, the existing pre-condition nodes representing previously asked queries in the preference network that match the current query will be set as evidence. Each interest concept from the current interest set is added to the preference network as a pre-condition node and set as evidence. If the user's query is totally new and the preference network is empty, the tool being used by the user is set to the default value (a filter) and a goal node representing the filter tool is added to the preference network. Otherwise, it is set to the tool being represented by the goal node with highest marginal probability. Each action node represents a way to construct a modified query based on the current tool, interests and user query.

In our user model, the preference network adapts based on the observation of interactions between a user and our system. The idea is based on Brown's work on interface agents (Brown et al., 1998). In our current version, there are two different tools, filter and expander. Therefore, two new preference networks are created, one of them contains an additional new tool labeled as filter, and another contains a new tool labeled as expander. We will then calculate the probability that a new network will improve the user's effectiveness for both of the two new

preference networks. The calculation is simple, which is to find out the frequency that a tool helps in the previous retrieval process. Currently, if the total number of retrieved relevant documents excesses a user-defined threshold, the tool used for the query modification is considered as helpful. The preference network updates itself according to the one with higher probability.

In the example shown in Figure 5, those nodes labeled "*transfer_rate_distribution*", "*zero_pressure_surface*", and "*heat_transfer*" are pre-condition nodes created from a user's current interest list while the nodes "*query_01*" and "*query_02*" nodes are pre-condition nodes created to represent a user's queries. The nodes "*filter_01*" and "*expander_01*" are two goal nodes and "*proactive_query_01*" and "proactive_query_02" are action nodes. After we set as evidences three nodes labeled "*transfer_rate_distribution*", "*query_01*", and "*zero_pressure_surface*" (shown as shaded nodes in the Figure 5), we can perform belief updating on this preference network. The node "*filter_01*" has value 1 for state "true" and 0 for false while node "expander" has value 0.42 for state "true" and 0.58 for state "false", and therefore, we will use the filter to modify the original query.

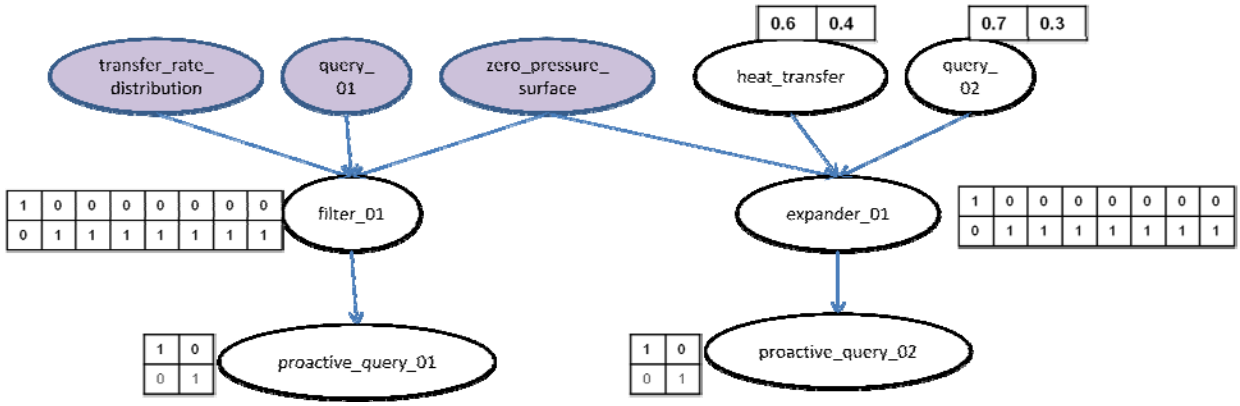



**Figure 5**: An example of a preference network.

**Construction of modified query graph**

The procedure for constructing a modified query graph from the current context network, interest set, preference network and the user's original query graph is as follows:

- Given a user model *M={I, P, C}* and a query graph *q*. *I* is an interest set. *P* is a preference network which is a Bayesian network to reason about the tool used for modifying a query. *C* is a context network which is a directed acyclic graph containing concepts and relation nodes.
- Use spreading activation algorithm to reason about the new set of interest *I'*.
- Set as evidence all concepts of the interest set *I'* found in the preference network *P*.
- Find a pre-condition node a representing a query in the preference network P which has associated QG that completely or partially matches against the given query graph *q*. If such a node *a* is found, set it as an evidence.
- Perform belief updating on the preference network P. Choose top *n* goal nodes from preference network with highest marginal probability values. Call this set of goals as suggested goal nodes.

– For every goal node in the set of suggested goal nodes, do

  If the query has been asked before and the user has used this goal, 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 the goal node 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 the context network $C$. For every concept $i$ in $I'$, we search for its corresponding node $ci_i$ in the context network such that $ci_i$ is an ancestor of $cq_i$. If such $ci_i$ and $cq_i$ are found, we add the paths from context network between these two nodes to the modified query graph.

  If the query has not been asked before and the goal node represents an expander: For every concept node $q_i$ in the user's query graph $q$, we search for its corresponding node $cq_i$ in the context network C. For every concept $i$ in $I'$, we search for its corresponding node $ci_i$ in the context network such that $ci_i$ is a progeny of $cq_i$. If such $ci_i$ and $cq_i$ are found, we add the paths from context network between these two nodes to the modified query graph.

Figure 6 shows an example of an original query graph. In this example, we assume that the query has not been asked and currently the goal node fired in the preference network representing a filter. Furthermore, the concept "*transfer_rate_distribution*" is currently in the interest list. The current context network contains the following relations:

transfer_rate_distribution → related to → transfer

transfer_rate_distribution → related to → zero_pressure_surface → related to → surface

In the modified query graph in Figure 7, the concept nodes representing "*transfer rate distribution*", and "*zero pressure surface*", and their relations with the concepts "*transfer*" and "*surface*" are newly added. The relation "transfer_rate_distribution → related to → transfer" is added first because the node "*transfer_rate_distribution*" is found both in the interest list and context network and is an ancestor of the node "*transfer*" in the query graph. Similarly, the relation "*transfer_rate_distribution → related to → zero_pressure_surface → related to → surface*" is added.

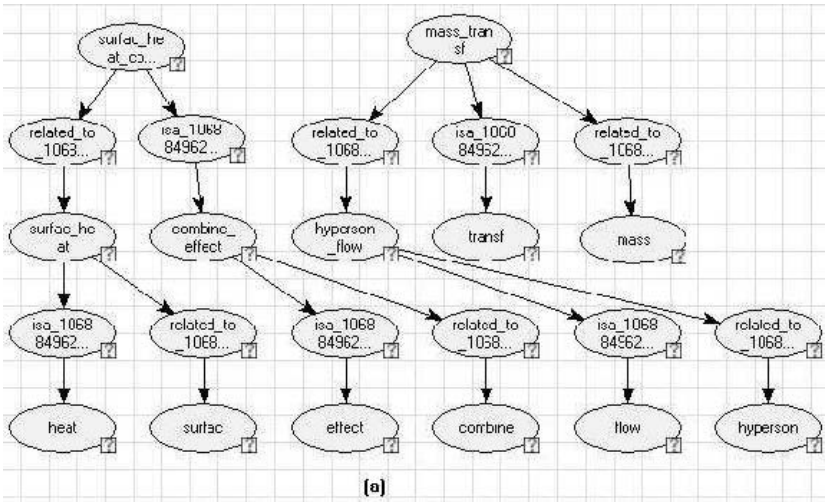



**Figure 6**. A user's original query graph

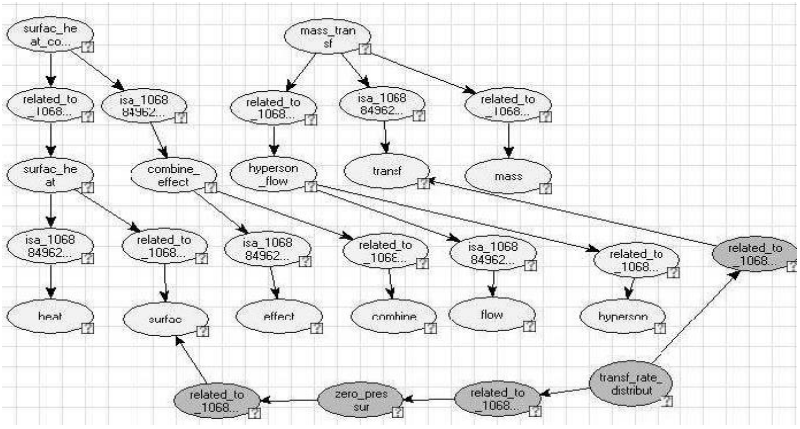



**Figure 7**. A modified query graph

**Integrating IPC into information retrieval application**

In our approach, we generate a document graph for each document automatically in an offline process. In this process, we use LinkParser Program (Sleator & Temperley, 1993) to parse each sentence in a document and extract noun phrases, prepositional phrases from a parsed tree to construct a document graph using three heuristics listed in Appendix. Even though LinkParser package comes with the capability with a full parser, we mainly use it for recognizing the noun phrases and prepositional phrases. The running time for LinkParser is $O(l^3)$ for each sentence in which $l$ is the number of words in the sentence to be parsed. The complexity of the process of creating relations from the phrases extracted from the Link Parser is $O(l \log l)$. We generate a query graph for each query in the same way as we generate a document graph. Therefore, we always maintain the consistency between a query graph and a document graph. While we are trying to maximize the quality of document graphs as much as possible, the consistency between query graph and document graph is very important to ensure good retrieval performance. For constructing Bayesian networks, we use Smile package from the Decision Systems Laboratory, University of Pittsburgh, to implement our preference network (Druzdzel, 1999).

The architecture of our IR application includes five modules as shown in Figure 8. The user model module takes as input a query graph and generates as output a modified query graph for the search module. The search module matches the modified query graph against each document graph representing a record in the database of documents, chooses those records that have the number of matches greater than a user-defined threshold, and displays the output to the user. A match between a QG $q$ and a DG $d_i$ is defined as:

$$sim(q, d_i) = \frac{c}{2 * C} + \frac{r}{2 * R}$$

in which $c$ and $r$ are the number of concepts and relation nodes of the query graph found in the document graph while $C$ and $R$ are the total numbers of concept and relation nodes of the query graph. Note that two relation nodes are matched if and only if its parents and its child are matched. In this similarity, we treat relation nodes and concept nodes equally. After the search module returns the search results, the feedback module allows a user to indicate whether the search result is relevant or not. The document graphs of relevant documents will be used to update the three component of the user model as describe earlier.

In our approach, we use some thresholds such as for filtering out concepts in the short-term

interest lists and thresholds for finding common sub-graphs. These thresholds are determined based on empirical tests with the above collections.

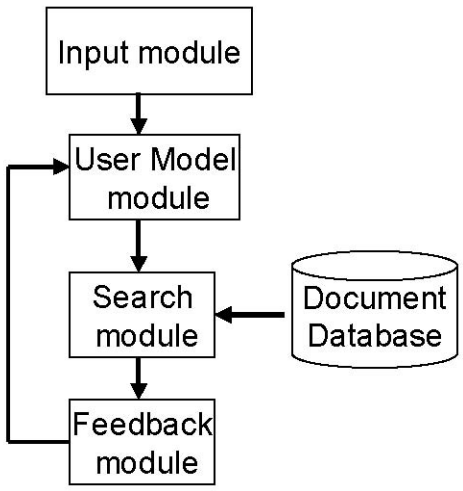



**Figure 8.** Overall architecture of IR application using our user model

**Analysis of user model complexity**

We now focus on the running time cost of our approach.

Construction/Update: Denote $n$ as the size of a document graph with the largest number of nodes of all the retrieved relevant documents. Denote $m$ as the number of retrieved relevant documents. The complexity of the construction of an interest set is $O(n+nm)$.The complexity of the spreading activation propagation algorithm to construct an adaptive interest set is $O(|C|^3)$ where $|C|$ is the size of the context network $C$. In our implementation, we have used a hash table to index the nodes in a document graph. The running time complexity of finding a set of common subgraphs in the intersections of $m$ document graphs is $O(m^2 n+mn)$.

The complexity of construction our preference network is $O((|I|+ |q|^2)|P|)$ where $q$ is the given query graph.

Inference: The complexity of inference on C is $O(|C|)$. The complexity of inference on $P$ is polynomial because we are using an inference algorithm for poly-trees (Pearl, 1988).

## Empirical Evaluation with Synthesized data set

The main goals of this empirical evaluation are to assess the effectiveness and competitiveness of our user model with regards to retrieval of relevant documents. We view the effectiveness of a user model as its ability to produce a desired effect when integrated with an IR application. Specifically, the desired effect in our case is to improve the number of retrieved relevant documents early in the process of searching for information. We aim at a unified set of empirical evaluations which offers an opportunity to compare our user modeling approach with existing approaches using relevance feedback from the IR community. We begin with a description of our testbed, present the technique we are comparing against, describe the experimental procedure, and finally present our results and discussion.

## Testbed

Our evaluation uses the CRANFIELD (Cleverdon, 1967), MEDLINE and CACM collections (Salton and & Buckley, 1990). CRANFIELD contains 1398 papers on aerodynamics and 225 queries with relevancy assessment. A relevant document with regards to a specific query is rated on a numerical scale from 1 to 4. 1 represents the most valuable reference and 4 represents the least valuable reference while -1 represents a document that is irrelevant. CACM contains 3204 documents and 64 queries in computer science and engineering (CSE) domain while MEDLINE contains 1033 documents and 30 queries in the medicine domain. Our main reason for choosing these collections is that they have been used for evaluating the effectiveness of techniques that use relevance feedback and query expansion (Lóper-Pujalte et al., 2003; Salton & Buckley, 1990). Additionally, the sizes of these collections seem to be appropriate for a user modeling application to start with.

For CRANFIELD, we choose this set of 43 queries with the property that at least 6 relevant documents are not in the top 15. For MEDLINE and CACM, we use the whole query set.

In our experiment, we use both original and residual collections. Original is referred to the document set obtained when the same query is run again without removing the relevant document retrieved when this query is issued for the first time. Residual collection is created by removing all documents from top 15 in the initial run regardless to whether or not they are relevant or not from the original collection.

## TFIDF and Ide dec-hi approach

We compare the effectiveness of our user model approach against Ide dec-hi with term weighting using TFIDF (Salton & Buckley, 1990). There are several reasons for this. First of all, TFIDF and Ide dec-hi techniques are very well-documented (Frake & Baeza-Yates, 1992; Lóper-Pujalte et al., 2003; Salton & Buckley, 1990), and thus make it easier and more reliable for re-implementation. Secondly, the Ide dec-hi approach was considered to be the best traditional approach in the IR community and is still competitive with other recent approaches (Drucker et al., 2002; Lóper-Pujalte et al., 2003; Salton & Buckley, 1990). It offers an opportunity to see where we stand with other approaches as well because others also report their results comparing against Ide dec-hi.

To briefly summarize, the differences between our approach and Ide dec-hi using TFIDF are three-fold. First, we capture the structure of information instead of frequency of individual terms. Second, our approach determines a user's intent in information seeking to decide which concepts and relations to add to the original query instead of adding the terms directly from relevant and non-relevant documents to a user's original query. Lastly, in our approach, the information learned from relevance feedback for a given query can be reused for the same or related query while it is only used for the same query in Ide dec-hi.

We re-implement TFIDF and Ide dec-hi as described in (Salton & Buckley, 1990) using the vector space model. A query is processed in the exact procedure as we process a document. The similarity between a document vector and a query vector is defined as the cosine between them as shown in (Frake & Baeza-Yates, 1992, chapter 14, page 366).

The main idea of Ide dec-hi is to merge the relevant document vectors into the original query vector. This technique automatically reweighs the original weight for each term in the query vector by adding its corresponding weights from relevant documents directly and subtracting its corresponding weight from the first non-relevant document. For the terms which are not from the

original query vector but appear in the relevant documents, they are added automatically to the original query vector with their associated weights. For the terms which are not from the original query vector but appear both in the relevant document and non-relevant documents, their weight would be the difference between the total weights of all relevant documents and the weight in the first non-relevant document. For the terms which are not from the original query vector but appear only in non-relevant documents, they are not added to the original queries vector with negative weights (Frake & Baeza-Yates, 1992). The formula for Ide dec-hi is:

$$Q_{new} = Q_{old} + \sum_i D_i - D_j$$

in which $Q_{new}$ and $Q_{old}$ represent the weighting vector for the modified query and the original query, respectively. $D_i$ represents the weighting vector for any relevant document and $D_j$ represents the weighting vector for the first non-relevant document. Denote $n$ as the size of the biggest vector representing a relevant document. Denote $m$ as the number of relevant documents. The running time for Ide dec-hi approach is $O(m*n)$.

**Traditional procedure applied to Ide dec-hi/TFIDF and user modeling**

We simulate the procedure laid out by Salton and Buckley (1990). For the Ide dec-hi/TFIDF, each query in the testbed is converted to a query vector. The query vector is compared against each document vector in the collection. For our approach, we construct a QG for each query in the testbed. After we issue each query, the relevant documents found in the first 15 returned documents are used to modify the original query. For the Ide dec-hi/TFIDF, the weight of each term in the original query is modified from its weights in relevant documents and the first non-relevant document. The terms with the highest weights from relevant documents are also added to the original query. For our 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 Section3. We then run each system again with the modified query. We refer to the first run as *initial run* and the second run as *feedback run*. After the feedback run, we reset the user model to empty. For each query, we compute average precision at three point fixed recall (0.25, 0.5, and 0.75).

**New procedure for user modeling approach**

The new procedure is similar to the traditional procedure described earlier except that we start with an empty user model in one experiment and with a *seed* user model in some experiments. The seed user model is the model that is created after we run the system through the set of queries once. For each query, we still use the same set of documents. The new procedure assesses the effects of prior knowledge and the combination of prior knowledge with knowledge learned from a query or a group of queries. We would like to separate what we term "short-term" and "long-term" effects. Short-term effects are the effects obtained by using the user model to modify the same query immediately after a user is giving feedback. Long-term effects are the effects obtained by using user model to modify any queries regardless of whether feedback has been given or not. Our goals for this new procedure are to evaluate:

- How does the long-term effect of our user model affect the retrieval performance?
- How do the combination of short-term and the use of prior knowledge affect the retrieval performance?

- How do the combination of short-term, long-term and the use of prior knowledge affect the retrieval performance?

These requirements lead to our decision to perform 4 experiments:

Experiment 1: We start with an empty user model. We update the initial user model based on relevance feedback and we do not reset our user model unlike the traditional procedure above. The user model obtained at the end of this experiment is used as the seed user model for the next 3 experiments.

Experiment 2: We start with the seed user model. For each query, we do not update our user model and don't run the feedback run. This experiment assesses how the prior knowledge helped improve retrieval performance.

Experiment 3: We start with the seed user model and run our system following the traditional procedure described above. However, after each query, we reset our user model to the seed user model. This experiment assesses the effects of the combination of prior knowledge and knowledge learned from a given query on retrieval performance.

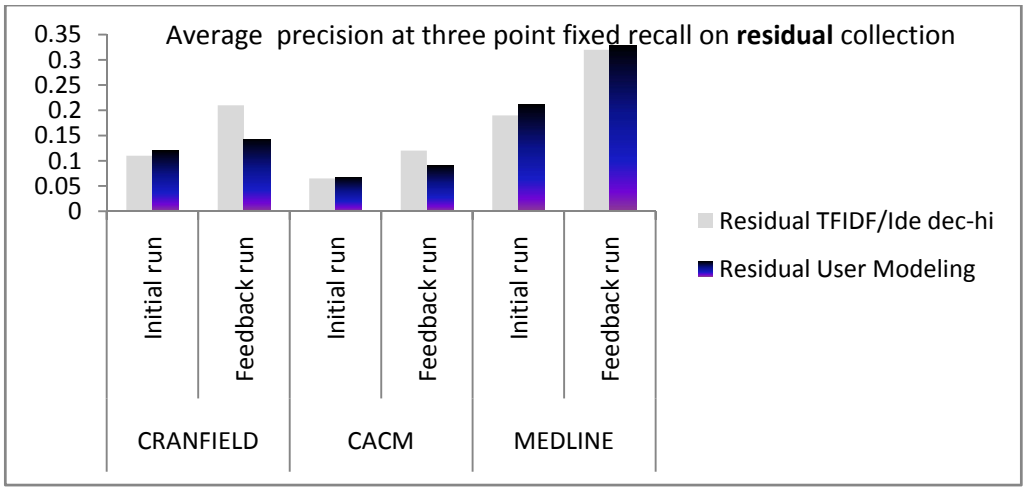
Experiment 4: We start with the seed user model. For each query, we update our user model based on relevance feedback and we do not reset our user model. This experiment assesses the effects of combination of prior knowledge, and knowledge learned immediately from each query and knowledge learned from previous queries on retrieval performance.

The modified query graph from all experiments will be matched against every document graph in the database. We compute the similarity based on the formula previously described. We return every document which has a similarity greater than zero.

For all experiments in the traditional and new procedure, except experiment 2, we compute average precision at three point fixed recall (0.25, 0.5, and 0.75) for both initial run and feedback run using original collection and residual collection. For Experiments 2 in the new procedure, we compute the average precision at three point fixed recall for only the initial run.

**Results**

The result of the traditional evaluation of Ide dec-hi using TFIDF and our user modeling approach is shown in Figure 9. For CRANFIELD collection, the precision of the initial run and feedback run using residual collection are close to the numbers reported by (Lóper-Pujalte et al., 2003) and (Salton & Buckley, 1990). Those in previous publications for CACM and MEDLINE achieved a slightly better results compared to ours because (i) we used the entire set of queries, while others, for example (Lóper-Pujalte et al., 2003), used a subset of queries; and (ii) we treat the terms from title, author, and content equally. Figure 9 shows that we achieved competitive performance in both runs for residual and original collections for MEDLINE collection. For CRANFIELD and CACM collections, the Ide dec-hi approach obtained higher precision in the feedback run with residual collection. However, we still obtained competitive results for both two collections in the feedback run with original collection. The reason is further investigated and presented in more details in the discussion section.



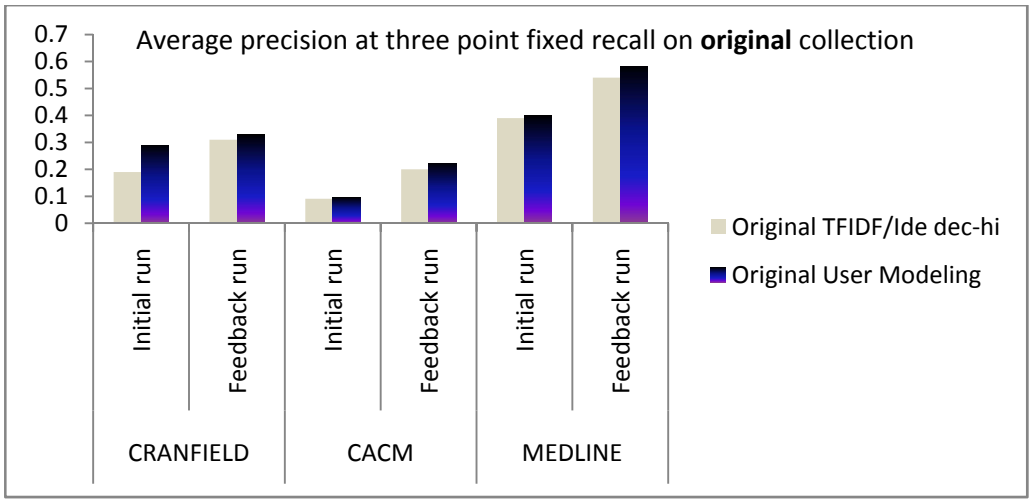




**Figure 9:** Average precision at three point fixed recall for traditional procedure

The results of our new procedure evaluation are reported in Figure 10.

Results of Experiment 1: The first experiment also shows that the user model did improve the retrieval performance in the feedback run compared to the initial run for all three collections.

Results of Experiment 2: The precision of this experiment is higher than the experiment in the traditional procedure for CRANFIELD and MEDLINE collections. That said, the seed user model is doing its job by changing the user's queries and improves the retrieval performance in the initial run.

Results of Experiment 3: The precision of this experiment, as shown in Figure 10, indicates that there is a relatively good improvement of the feedback run compared to the initial run on residual collection for CRANFIELD and MEDLINE. This experiment shows that the more knowledge a user model has about a user and search domain, the better it helps improve retrieval performance.

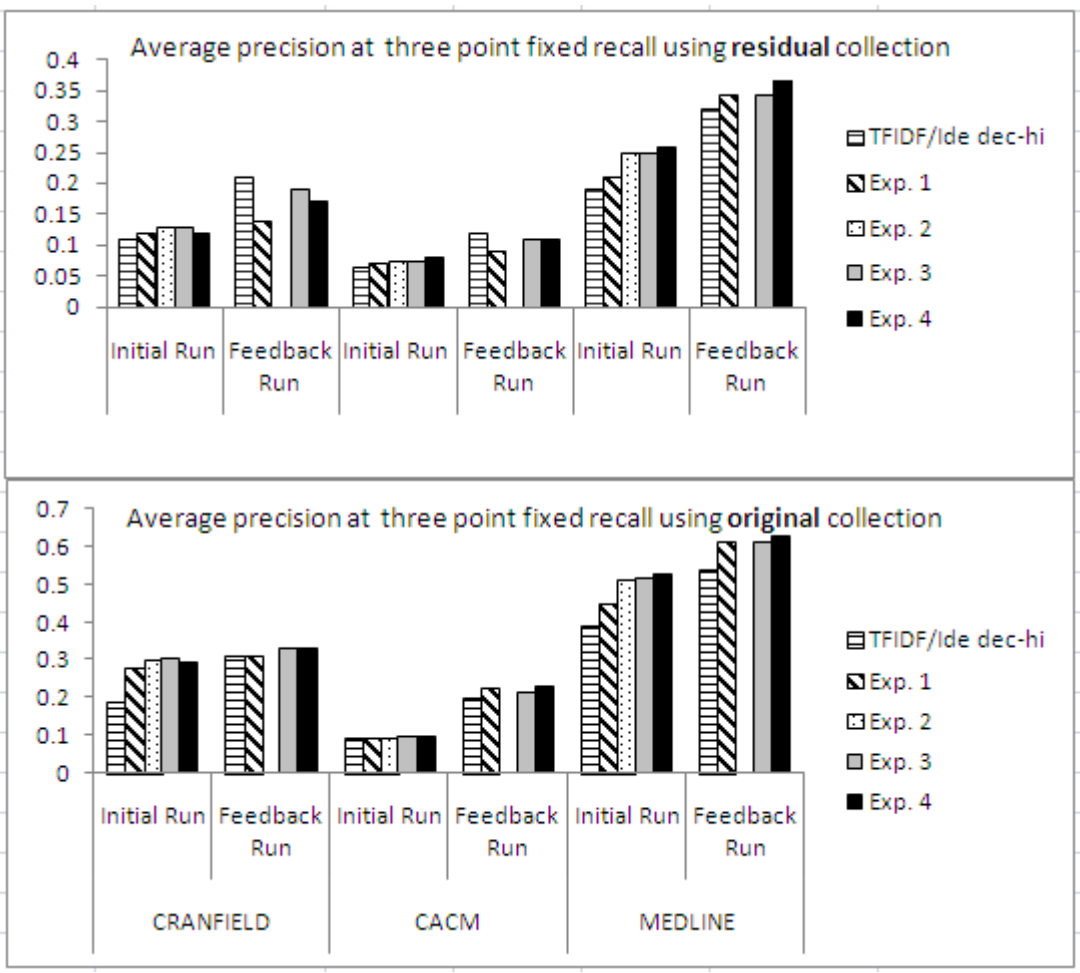



**Figure 10:** Average precision at three point fixed recall for new procedure

Results of Experiment 4: For CRANFIELD collection, the precision of the feedback run using residual collection is slightly less than the precision of Experiment 3, but is better than the precision of traditional procedure and Experiment 1. For both CACM and MEDLINE, we can see that among the four experiments, Experiment 4 performs competitively compared to Ide dec-hi in the feedback run while it offers the advantages of having higher precision in the initial run compared to TFIDF. The reason for this is that within the design of this experiment, we use the same seed user model in Experiments 2 and 3, the database is the same, the query set is the same and the set of relevant documents used in updating the user model is taken from top 15 of retrieved documents for every query. Probably, there is no new information learned by the user model, therefore we can't expect Experiment 4 to give better results. However, in the context of an actual interactive IR framework where the database may change overtime, users give different feedback over time; as such our user model should help target IR application achieve much better results. We are able to find significant effect in the initial run (with original collection) for CRANFIELD collection (p-value $< 0.05$) but we are unable to find significant effect with feedback runs for CRANFIELD collection, as well as with initial runs with CACM and MEDLINE (p-value $> 0.05$).

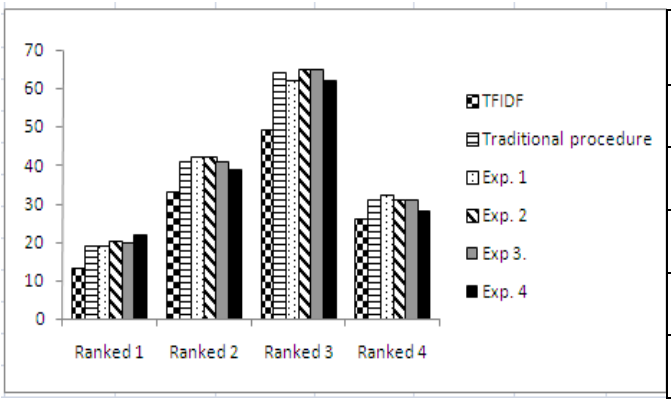
**Discussion**

Our goals initially set for these experiments have been met.

- Effectiveness of the user model in terms of improving the retrieval performance: Experiments 1, 3, and 4 show that by using our user model, the precision of the feedback run is always higher using residual and original collections compared to the initial run.

- Competitiveness of the improvement obtained by using our user model compared with the improvement obtained by using TFIDF and Ide dec-hi approach: Among two procedures and 5 experiments in total, we can see that for CRANFIELD Experiment 3 of new procedure performed competitively well compared to Ide dec-hi while it offers the advantages of having higher precision in the initial run compared to TFIDF. For both CACM and MEDLINE, the Experiment 4 of new procedure performs competitively with Ide dec-hi in the feedback run.

Moreover, as the CRANFIELD collection has classified relevancy of a document on a 4-point numeric scale for all relevant documents, we investigated further to see how many good relevant documents our approach has retrieved from the initial runs compared to TFIDF. We found out that the number of relevant documents in the top 15 of the initial runs in all six experiments using our approach for all queries in the testbed is always higher than that of the initial runs using TFIDF (as shown in the last row of Figure 11). Figure 11 shows that all of our experiments retrieved more documents ranked 1 than the TFIDF approach, ranging from 46% to 69%. We are able to find significant effect using t-test on the number of retrieved relevant documents found in top 15 (as shown in Figure 11). We also retrieved more relevant documents in all ranks in top 15. This indicates that the retrieval quality of our approach is significantly better. As a result, there are less relevant documents, especially, the most valuable relevant documents, left to be retrieved in the feedback run. This is the main reason why the improvement of precision in the feedback run on the residual collection for our approach is less than that achieved by Ide dec-hi approach.




| Degree of freedom | Experiment | t, p-value |
|---|---|---|
| 3 | Traditional vs. TFIDF/Ide dec-hi | t=3.77 p-value=0.033 |
| 3 | Exp. 1 vs. TFIDF/Ide dec-hi | t= 5.13 p-value=0.014 |
| 3 | Exp. 2 vs. TFIDF/Ide dec-hi | t=3.86 p-value=0.031 |
| 3 | Exp. 3 vs. TFIDF/Ide dec-hi | t=3.73 p-value=0.034 |
| 3 | Exp. 4 vs. TFIDF/Ide dec-hi | t=3.22 p-value = 0.048 |

**Figure 11.** Retrieved relevant documents in top 15 for CRANFIELD and t-test results.

Regarding the evaluation methodology, the traditional procedure offers us a chance to compare with the TFIDF and Ide dec-hi approaches using their evaluation procedures on the same collections. For any user modeling approach, the ultimate goal is to test the approach with a group of end users. This test however, often times is considered as expensive and contains many variables to control. Our evaluation methodology helps us to prepare better for the test with end users because the testbed contains knowledge spreading over several domains and the tested queries are as complicated as the ones asked by any end user. The evaluation procedure fortunately is light-weight and they can be easily used to evaluate adaptive systems before hiring the real

subjects. This maintains objectivity and serves as a baseline comparison for future extensions.

There are four main conclusions that we have after conducting our experiments:

- Our approach of capturing user intent in information seeking and using the information captured to modify user queries does produce better performance in the initial runs compared to TFIDF. The quality of retrieval in our initial run is significantly better than TFIDF for CRANFIELD collection.
- Our user model is trying to combine and balance between the long term and short-term interests of a user in information seeking. The model is long-lived, therefore, even its long-term effects can help modify queries and improve retrieval performance in a search session.
- We show how we can compare the user modeling approaches using procedures, collections and metrics of the IR community while still being able to assess special features of the models such as the use of prior knowledge, knowledge learned from one query or from a group of queries.

## Evaluation with human analysts

After our evaluation with synthesized data sets, we had an opportunity to conduct an evaluation with three human analysts (Santos et al., 2005). The evaluation took place at a laboratory of the National Institute of Standards and Technology (NIST) in May, 2004. This system is compared with commercial off-the-shelf keyword based application called Verity Query Language (VQL). We use a data collection from the Center for Nonproliferation Studies (CNS, Sept. 2003 distribution. http://cns.miis.edu/). The UM system package, which includes the pre-processed CNS database with 4000 documents on weapons of mass destruction (WMD) domain, was delivered to and installed at the NIST laboratory. Three evaluators, who are naval reservists currently assigned to NIST with intelligence analysis background participated in the experiments. During the evaluation, the UM system and the VQL system were run side by side. The same queries were input into both systems and the retrieved documents were compared. For the VQL system, analysts needed to note on paper which documents were relevant to their interests for each query; for the UM system, in addition to recording the relevancy, they were asked to mark check boxes beside the documents if they were relevant ones. There was a short tutorial session to show the analysts how to work with the UM system, such as indicating the relevancy. The VQL system has a graphic user interface (GUI) similar to Google, thus, it is straightforward to use.

### Procedure

The experimental session lasted about 4 hours. Each participant was asked to perform a search on Qumar research and development of biological weapons. Note that the country name has been replaced. Because of the time constraint, the participants were asked to check the first 10 returned documents for relevancy only, and the task was limited with just10 fixed queries. We scripted 10 queries to avoid adding more variables into our experiments such as errors in natural language processing. The queries were extracted from the log of query sequences of working analysts and re-arranged from a database that collected other intelligence analysts IR activities at NIST laboratory.

The UM system started with an empty user model, which means that the user model initially knew nothing about the analyst, and had to start learning about the user from the very beginning. The procedure went as follows:

**Step 1**: each of the analysts was asked to fill out an entry questionnaire about their background and experience with searching programs; and, respond to an exit questionnaire about their experience on working with the UM system.

**Step 2:** start with the first query in the query list, each analyst starts both systems with the same query.

**Step 3**: after the system with user model approach returns a set of documents, each analyst is asked to look at the first 10 returned documents and indicate which one is relevant to the current query. Each analyst only sees the same set of retrieved documents for the first query but after that depending on which retrieved documents are marked as relevant, each analyst's model will be built and used to proactively modify subsequent queries. Therefore, the set of retrieved documents after the first query may be different for these three analysts.

**Step 4**: after deciding relevant documents, each analyst clicks on "Put Feedback" option.

**Step 5**: Start a new query for both systems.

**Step 6:** After finishing all queries, each analyst is asked to fill out the exit questionnaire about the workload spent.

**Results and Analysis**

The experience in intelligence analysis for the three participants ranged from five months to seven years. Two of them use computers as a tool in their analysis work, while one does not. They all felt comfortable with using search tools like Google, and considered themselves well-informed on the topics of WMD and terrorism. The most interesting observation is that the three analysts tended to take different approaches in searching for information. Analyst 2 looks at the big picture first; while Analyst 3 likes to start with the details. Analyst 1 practices a mixed approach that depends on his knowledge of the topic. If much was already known, then he would try to create an outline of the useful information; otherwise, he would look for some details first. After 4 hours, two analysts finished the 10 queries that we provided, and Analyst 3 finished 9 queries. All of them managed to identify more relevant documents when working with the UM system than they did with the VQL system. The precisions were 0.257 and 0.312 for the VQL system and the UM system, respectively. Since a document could be returned and identified multiple times as relevant for different queries, we also counted the numbers of unique (or distinct) documents that have been returned by the system and found as relevant by each participant. The data showed that when they were using the UM system, each of them was presented with more unique documents, and selected more unique documents as relevant as shown in Table I. The total number of unique relevant documents for all 10 queries returned by the UM system is 39, while the number is 27 by the VQL system, a 44% increase. The number of documents selected as relevant by more than 2 analysts are 15 in the UM system and 19 in the VQL system, respectively. Notice that the number of documents marked as relevant by just one analyst is 24 when using the UM system, while this number is only 12 for the VQL system. This suggests that more information that is specifically relevant to each analyst's individual interests had been retrieved by the UM system. By using the UM system, the analysts displayed their differences in identifying the documents that were relevant to their individualized interests and searching styles. By the end of the experiment, the analysts were asked to fill out the exit questionnaire. When asked about the system performance and their satisfaction, they scored the UM system as above medium (3.7/5.0) (as shown in table II).

Notice that they felt the UM system was somewhat demanding, especially in mental effort and the temporal effort. Since relevancy assessment is a mentally demanding process by itself, and the analysts were required to finish the experiment in about 4 hours, which included 10 queries (i.e., more than 100 documents to review, of which some of them may be quite long), and working with 2 different systems at the same time, we think this is a result of the workload the analysts had in the experiments. As the data shows, the UM system presented more unique documents to the analysts, and helped analysts retrieve more relevant documents. In particular, it helped them retrieve more information that is relevant to their individual interests, which suggests that the user model was tracking the user's personalized interests.

| | **VQL** | **UM** |
|---|---|---|
| Total unique relevant documents | 27 | 39 |
| Document marked as relevant by all 3 | 3 | 8 |
| Documents marked as relevant by more than 2 analysts | 19 | 15 |
| Document marked as relevant by only one analyst | 12 | 24 |

**Table I**. Unique relevant documents retrieved by two systems.

| Questions | Score |
|---|---|
| How confident were you of your ability to use the system to accomplish the assigned task? (1-5, 1: less confident, 5 more confident) | 3.0 |
| Given that you were performing this task outside of your standard work environment, without many of your standard resources, were you comfortable with the process of preparing your report? (1-5, 1: less comfortable, 5: more comfortable) | 3.7 |
| Given that you were performing this task outside of your standard work environment, with access to a restricted set of documents, were you satisfied with the quality of the report/answers that you were able to find for this scenario? (1-5, 1: not satisfied, 5: satisfied) | 2.7 |
| How satisfied are you with the overall results for this task using system with user model? (1-7. 1: most satisfied, 7: least satisfied) | 4.3 |
| How confidence are you with the results that they cover all possible aspects of the task? (1-7. 1: most confident, 7: least confident) | 4.7 |
| The regarding this task, do you think that user modeling approach helped you to retrieve critical document earlier in the process than the VQL? (1-7. 1: strongly agree, 7: strongly disagree) | 3.7 |
| Ranking of mental demand. (1-7, 1: little, 7: high) | 5.3 |
| Ranking of physical demand. (1-7, 1: little, 7: high) | 2.0 |
| Ranking of temporal demand. (1-7, 1: little, 7: high) | 5.0 |
| Ranking of performance demand. (1-7, 1: little, 7: high) | 4.7 |
| Ranking of frustration. (1-7: 1: little, 7: high) | 5.3 |
| Ranking of effort. (1-7. 1: little, 7: high) | 6.0 |

**Table II**. Average score for user satisfaction of the User Modeling approach.

## Conclusions and Future work

In this chapter, we have described the development and the evaluation of a user model to capture a user's intent in information seeking for improving retrieval performance. The difference between ours and the existing relevance feedback approaches in the IR community is that we capture the structure of information while existing approaches in IR focuses on frequency of individual terms.

Also, we use the information about a user's intent to guide the process of modifying the user's queries instead of merging individual terms directly from relevant and non-relevant documents into the user's original queries. Our user model is long-lived. This means that feedback information can be used for a specific query or any other queries on the same or related topics. We have shown that our approach offers better performance in the initial runs and competitive performance in the feedback runs. The results from our research show that a user model can be a very effective tool for improving retrieval performance in an IR interactive framework.

Our evaluation with three intelligent analysts partially answered the question on impacts of user modeling on an IR system by measuring the number of relevant documents presented to the analysts. This user modeling approach also tracks the individual differences among analysts and presents uniquely relevant documents to each analyst. By combining these results, we can assess if the user modeling is actually follows the user's individual interests, and ultimately improve the user's performance in an IR system.

There are issues that we wish to address from this research. First, our user model currently can only modify a user's query. We are looking at a methodology to allow the user model to modify other parameters and attributes of the target IR system in a decision theoretic framework. For example, the similarity measure or threshold can be adaptively changed based on a user's searching behaviors. We have pushed our effort further to develop a hybrid user model as an extension of the IPC model (Nguyen et al., 2006). Secondly, the process of constructing a model for a user is done automatically by the system using inputs from a user's query and reference feedback. Relevance feedback is a lightweight way to get a user's inputs and build a model. However, in the long run, a user may become frustrated because he/she did not know how the feedback is used. This is also pointed out in our experiment with human analysts. We plan to employ an explanation mechanism which uses context network, interest and preference network to provide natural language description of why a query is modified in a certain way. We also allow users to manually re-start the model when a new topic is issued and the model has not changed fast enough and allow users to manually edit context network at any time.

Next, our user modeling approach works best if a user has demonstrated his/her searching styles. So, we will consider re-ordering the queries to affect different search styles (e.g users explore a topic, its subtopics, and then change to a new topic). It will help to closely relate the experiment to real life situations while maintaining its objectivity. Fourth, we would like to explore some other semantic relationships such as "links to", "associated with", and "caused by" by using discourse analyses, heuristics such as (Greffenstette & Hearst, 1992). More specifically, we investigate the possibility to reason about these relationships given a sequence of actions, snippet and annotation associated with each action. Fifth, we would like to push our effort further in evaluation the effectiveness of our approach. We plan to extend our empirical evaluation on data from recent TREC conferences. The scalability issue should be taken into consideration while we are working with large scale testbeds. We would like to explore the possibilities of working with abstracts of large documents instead of considering entire documents and the use of the parallel computing to accelerating the process. Additionally, we want to strength our evaluation results on internal accuracy of our user model (Nguyen & Santos, 2007a) as well as the use of prior knowledge in our user modeling approach (Nguyen & Santos, 2007b). Finally, we are mapping our current approach to modeling context into Bayesian knowledge-bases (Santos & Santos, 1998; Santos & Dinh, 2008) to better utilize probabilistic semantics for uncertainty like its use in our preference network.

## Acknowledgement

This work has been supported in part by grants from the Advanced Research Development Activity, the Intelligence Advanced Research Projects Activity, the Air Force Office of Scientific Research, and the Air Force Research Laboratory. Special thanks to the anonymous reviewers whose comments helped to greatly improve this chapter.

## Appendix

The flowchart of the algorithm for extracting document graph from a natural language text file is shown in Figure 9.

1. Note that in this flowchart, we used Link Parser (Sleator & Temperley, 1993) as a tool to parse a natural language sentence.

2. The relations are extracted based on three heuristic rules which are *noun phrase heuristic*, *noun phrase-preposition phrase heuristic* and *sentence heuristic*. Noun phrase heuristic captures taxonomy relations within a noun phrase. For example: for the noun phrase of "*experimental investigation*", we have extracted the relation: *experimental investigation – isa – investigation*.

   Noun phrase-preposition phrase heuristic attaches prepositional phrases to adjacent noun phrases. For example: "*boundary layer transition at supersonic speeds"*. We have extracted the relation: *boundary layer transition -related to -supersonic speeds*.

   Sentence heuristic relates two noun phrases associated with a verb in a sentence. For example: "*the discussion here is restricted to two-dimensional incompressible steady flow*". We have extracted the relation: *discussion - related to - two dimensional incompressible steady flow.*

3. We use "related to" to represent relations in a conjunctive noun phrase. For example: with the phrase "*heat transfer and zero pressure transfer*", we extracted the relation "*heat transfer - related to - zero pressure transfer*". We do not have any special rules for pronouns currently.

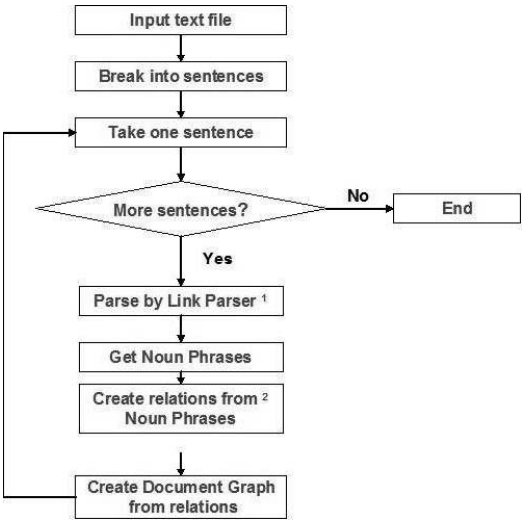



**Figure 12.** Flowchart of the algorithm to extracting document graph

## References


Allen, R. (1990). User Models: theory, method and practice. *International Journal of Man-Machine studies*, 32, 511–543.

Baeza-Yates, R., Calderón-Benavides, L., & Gonzalez-Caro, C. (2006). The Intention Behind Web Queries. In *Proceedings of String Processing and Information Retrieval 2006* (pp 98-109). Glasgow, Scotland.

Balabanovic, M. & Shoham, Y. (1997). Content-based collaborative recommendation. *Communications of the ACM, 40(3),* 66–72.

Balabanovic, M. (1998). Exploring versus Exploiting when Learning User Models for Text Recommendation. *Journal of User Modeling and User-Adapted Interaction, 8(1)*, 71–102.

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

Billsus, D. & Pazzani P. 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.

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

Broder, A. (2002). A taxonomy of Web Search. In *SIGIR Forum 36(2).*

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

Brown, S. M., Santos, E. Jr., Banks, S. B., & Oxley, M. (1998). Using explicit requirements and metrics for interface agent user model construction. In *Proceedings of the Second International Conference on Autonomous Agents* (pp. 1–7).

Brusilovsky, P. & Tasso, C. (2004). Preface to Special Issue on User Modeling for Web Information Retrieval. *User Modeling and User-Adapted Interaction*, *14(2-3)*, 147-157.

Bueno, D. and David A. A. (2001). METIORE: A personalized information retrieval system. In Bauer, M., Vassileva, J. and Gmytrasiewicz, P. (Eds.). *User Modeling: Proceedings of the Eight International Conference, UM 2001,* (pp. 168–177).

Campbell, D. R., Culley, S. J., Mcmahon, C. A., Sellini F. (2007). An approach for the capture of context-dependent document relationships extracted from Bayesian analysis of users' interactions with information. *Information Retrieval,* 10(2), (Apr. 2007), 115-141.

Campbell, I. (1999). Interactive Evaluation of the Ostensive Model, using a new Test-Collection of Images with Multiple Relevance Assessments. *Information Retrieval 2(1),* 89-114.

Case, D. (2002). *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior.* Academic Press

Chin, D. (1989). KNOME: Modeling What the User knows in UC. In A. Kobsa and W. Wahlster (Ed.), *User models in dialog systems,* (pp. 74—107). Springer Verlag, Berlin.

Chin, D. (2001). Empirical Evaluation of User Models and User-Adapted Systems. *User Modeling and User-Adapted Interaction 11(1-2),* 181-194.

Chin, D. (2003). Evaluating the Effectiveness of User Models by Experiments. *Tutorial presented at the Ninth International Conference on User Modeling (UM 2003).* Johnstown, PA

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

Cool, C. & Spink, A. (2002). Issues of context in information retrieval (IR): an introduction to the special issue. *Information Processing Management 38(5)* (Sep. 2002), 605-611.

Craswell, N., Hawking D., Upstill, T., McLean, A., Wilkinson, R., & Wu, M. (2003). TREC 12 Web and Interactive Tracks at CSIRO. *NIST Special Publication 500-255. The Twelfth Text Retrieval Conference,* (pp 193-203).

Crestani, F., Ruthven, I. (2007). Introduction to special issue on contextual information retrieval systems. *Information Retrieval, 10 (2) (Apr. 2007),* 111–113.

Decampos, L., Fernandez-Luna, J., & Huete, J. (1998). Query expansion in information retrieval systems using a Bayesian network-based thesaurus. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, (pp. 53–60). Sanfrancisco, CA.

deCristo, M. A. P., Calado, P. P., da Silveria, M. L., Silva, I., Munzt, R. & Ribeiro-Neto, B. (2003). Bayesian belief networks for IR. *International Journal of Approximate Reasoning 34,* 163–179.

Dervin, B. (1997). Given a context by any other name: methodological tools for taming the unruly beast. In P.Vakkari, R.Savolainen, & B.Dervin (Eds.), *Information seeking in context: Proceedings of an international conference on research in information needs, seeking and use in different contexts,* (pp.13–38). London: Taylor Graham.

DeWitt, R. (1995). Vagueness, Semantics, and the Language of Thought. *Psyche,* 1. Available at http://psyche.cs.monash.edu.au/index.html.

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

Druzdzel, J. M. (1999). SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A development environment for graphical decision-theoretic models (Intelligent Systems Demonstration). In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99),* (pp. 902-903), AAAI Press/The MIT Press, Menlo Park, CA.

Efthimis, E. N. (1996). Query Expansion. In Williams, M. (Ed.). *Annual Review of Information Science and Technology 31*, 121–187.

Frake, W. B. & Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice Hall PTR, Upper Saddle River, NJ 07458.

Goh, D. & Foo, S. (2007). *Social Information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively*. Premier Reference Source.

Greenberg, S. & Witten, I. (1985). Adaptive personalized interfaces - A question of viability. *Behaviour and Information Technology 4(1),* 31-45

Greffenstette, G., & Hearst, M. A. (1992). A method for refining automatically-discovered lexical relations: Combining weak techniques for stronger results. In *Proceedings of the Workshop on Statistically-Based Natural Language Programming Techniques,* AAAI Press, Menlo Park, CA.

Haines, D. & Croft W. B. (1993). Relevance feedback and inference networks. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA.* (pp. 2–11).

Hernandez, N., Mothe, J., Chrisment, C., & Egret, D. (2007). Modeling context through domain ontologies. *Information Retrieval,* 10 (2) (Apr. 2007), 143-172.

Horvitz, E., Breeze, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumiere project: Bayesian user modeling for inferring goals and needs of software users. In: *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence,* (pp. 256–265).

Hwang, C. H. (1999). Incompletely and imprecisely speaking: Using dynamic ontologies for

representing and retrieving information. *Knowledge Representation Meets Databases*, 14-20
Ide, E. (1971). New experiment in relevance feedback. In: *The Smart system-experiments in automatic documents processing,* (pp. 337–354).
Ingwersen, P. (1992). *Information Retrieval Interaction*. London, Taylor Graham.
Jansen B., Booth D., & Spink A. (2007). Determining the User Intent of Web Search Engine Queries. In *Proceedings of the International World Wide Web Conference*, (pp 1149-1150). Alberta, Canada.
Karamuftuoglu, M. (1998). Collaborative information retrieval: toward a social informatics view of IR interaction. *Journal of the American Society for Information Science, 49(12)*, 1070 -1080.
Lee U., Liu Z. & Cho J. (2005). Automatic identification of user goals in web search. In *Proceedings of the International World Wide Web Conference 2005*, (pp. 391–400), Chiba, Japan.
Liu, Z., Chu, W., W. (2007). Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Information Retrieval, 10 (2) (Apr. 2007),* 173 – 202.
Lóper-Pujalte, C., Guerrero-Bote, V., & Moya-Anegon, F. D. (2003). Genetic algorithms in relevance feedback: a second test and new contributions. *Information Processing and Management 39(5)*, 669–697.
Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM 37(7)*, 31–40.
Magnini, B. & Strapparava, C. (2001). Improving user modeling with content-based techniques. In: *Bauer,M., Vassileva, J., and Gmytrasiewicz, P. (Eds). User Modeling: Proceedings of the Eighth International Conference, UM 2001*. (pp. 74–83).
Murray, T. (1997). Expanding the knowledge acquisition bottleneck for intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, *8*, 222-232.
Mylonas, Ph. Vallet, D., Castells, P., Fernandez, M. and Avrithis, Y. 2008. Personalized information retrieval based on context and ontological knowledge. *Knowledge Engineering Review,* Cambridge University Press, 23(1), 73-100.
Nguyen, H. (2005). *Capturing User Intent for Information Retrieval*. Ph.D dissertation. University of Connecticut.
Nguyen, H., & Santos, E., Jr. (2007a). An Evaluation of the Accuracy of Capturing User Intent for Information Retrieval. In *Proceedings of the 2007 International Conference on Artificial Intelligence* (pp. 341-350). Las Vegas, NV.
Nguyen, H., & Santos, E., Jr. (2007b). Effects of prior knowledge on the effectiveness of a hybrid user model for information retrieval. In *Proceedings of the Homeland Security and Homeland Defense VI conference*. Vol. 6538. Orlando, FL. March 2007.
Nguyen, H., Santos, E. Jr., Zhao, Q., & Lee, C. (2004a). Evaluation of Effects on Retrieval Performance for an Adaptive User Model. In: *Adaptive Hypermedia 2004: Workshop Proceedings -Part I,* (pp. 193–202), Eindhoven, the Netherlands
Nguyen, H., Santos, E., Jr., Schuet, A., & Smith, N. (2006). Hybrid User Model for Information Retrieval. In *Technical Report of Modeling Others from Observations workshop at AAAI-2006 conference.*
Nguyen, H., Santos, E.J., Zhao, Q. & Wang, H. (2004b). Capturing User Intent for Information Retrieval. In: *Proceedings of the Human Factors and Ergonomics society 48th annual meeting.* (pp. 371–375), New Orleans, LA.
Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*.

Morgan Kaufmann, San Mateo, CA.
Rochio, J. J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing,* (pp. 313–323).
Rose D. & Levinson D. (2004). Understanding User Goals in Web search. In *Proceedings of the International World Wide Web Conference 2004*, (pp 13–19), New York, USA.
Ruthven, I. & M. Lalmas. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review, 18(2),* 95 – 145.
Salton, G. & Buckley, C. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science 41(4),* 288–297.
Salton, G., & McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.
Santos, E. Jr, Nguyen, H., Zhao, Q. & Pukinskis, E. (2003b). Empirical Evaluation of Adaptive User Modeling in a Medical Information Retrieval Application. In: *Proceedings of the ninth User Modeling Conference UM 2003*, (pp. 292–296). Johnstown. Pennsylvania.
Santos, E. Jr, Nguyen, H., Zhao, Q., & Wang, H. (2003a). User Modelling for Intent Prediction in Information Analysis. In: *Proceedings of the 47th Annual Meeting for the Human Factors and Ergonomincs Society (HFES-03)*, (pp. 1034–1038).
Santos, E. Jr. and Dinh, H. T. (2008). Automatic Knowledge Validation for Bayesian Knowledge Bases. *Data and Knowledge Engineering*, 64, 218-241.
Santos, E. Jr., Brown, S. M., Lejter, M., Ngai, G., Bank, S., & Stytz, M. R. (1999). Dynamic User Model Construction with BayesianNetworks for Intelligent Information Queries. In: *Proceedings of the 12th International FLAIRS Conference.* pp. 3–7. Orlando. FL.
Santos, E. Jr., Nguyen, H., & Brown, M., S. (2001). Kavanah: An active user interface Information Retrieval Application. In: *Proceedings of 2nd Asia-Pacific Conference on Intelligent Agent Technology*, (pp. 412–423).
Santos, E. Jr., Santos, E., S., & Shimony, S., E. (2003c). Implicitly Preserving Semantics During Incremental Knowledge Base Acquisition Under Uncertainty. *International Journal of Approximate Reasoning 33(1)*, 71-94.
Santos, E. Jr., Zhao, Q., Nguyen, H., Wang, H. (2005). Impacts of User Modeling on Personalization of Information Retrieval: An evaluation with human intelligence analysts. In S.Weibelzahl, A.Paramythis, and J. Masthoff (Eds.). *Proceedings of the Fourth Workshop on the Evaluation of Adaptive Systems, held in conjunction with the 10th International Conference on User Modeling (UM'05),* (pp 19-26).
Saracevic, T. (1996). Relevance reconsidered. In: *Ingwersen, P and Pors, P.O. eds. Proceedings of the Second International Conference on Conceptions of Library and Information Science: Integration in Perspective. Copenhagen: The Royal School of Librarianship*, (pp. 201–218).
Saracevic, T., Spink A., & Wu, M. (1997). Users and Intermediaries in Information Retrieval: What Are They Talking About? In *Proceedings of the 6th International Conference in User Modeling UM 97,* (pp. 43–54).
Shafer, G. (1976). *A Mathematical Theory of Evidence*, Princeton University Press.
Sleator, D. D. and Temperley, D. (1993). Parsing English with a link grammar. In: *Proceedings of the Third International Workshop on Parsing Technologies*, (pp. 277–292).
Spink, A. & Losee, R. M. (1996). Feedback in information retrieval. In Williams, M., (Ed.), *Annual Review of Information Science and Technology 31,* 33–78.
Weibelzahl, S. (2003). *Evaluation of Adaptive Systems*. Ph.D Dissertation. University of Trier, Germany.

Wilkinson, R. & Wu, M. (2004). Evaluation Experiments and Experience from Perspective of Interactive Information Retrieval. In *Adaptive Hypermedia 2004 - Workshop Proceedings -Part I. Eindhoven,* (pp 221-230), Eindhoven, the Netherlands.

Wilson, T. D. (1981). On user studies and information needs. *Journal of Documentation, 37(1)*, 3–15.

Zukerman I. and Albrecht, D. (2001). Predictive statistical models for user modling. In A. Kobsa (Ed.), *User Modeling and User-Adapted Interaction Journal*, 11(1-2), 5-18. Kluwer Academic Publishers.