Hybrid User Model for Capturing a User’s Information Seeking Intent

Eugene Santos

Smart Innovation, Systems and Technologies

Hybrid User Model for Capturing a User's Information Seeking Intent

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

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

Abstract: Modeling a user and their knowledge is a critical issue in successfully determining and evaluating collective intelligence. In this chapter, we study the problem of employing a cognitive user model for information retrieval in which knowledge about a user is captured and used for improving his/her performance in an information seeking task. This problem is very important to group intelligence analysis involving elements such as collaborative information retrieval and social information retrieval because knowledge about each individual needs to be used in order to improve a group's effectiveness in information seeking. Our solution is to improve the effectiveness of a user in a search by developing a *hybrid user model* to capture user intent dynamically and combines the captured intent with an awareness of the components of an information retrieval system. The term "hybrid" refers to the methodology of combining the understanding of a user with the insights into a system all unified within a decision theoretic framework. In this model, multi-attribute utility theory is used to evaluate values of the attributes describing a user's intent in combination with the attributes describing an information retrieval system. We use the existing research on predicting query performance and on determining dissemination thresholds to create functions to evaluate these selected attributes. This approach also offers fine-grained representation of the model and the ability to learn a user's knowledge dynamically. We compare this approach with the best traditional approach for relevance feedback in the information retrieval community - Ide dec-hi, using term frequency inverted document frequency (TFIDF) weighting on selected collections from the information retrieval community such as CRANFIELD, MEDLINE, and CACM. The evaluations with our hybrid model with these testbeds shows that this approach retrieves more relevant documents in the first 15 returned documents than the TFIDF approach for all three collections, as well as more relevant documents on MEDLINE and CRANFIELD in both initial and feedback runs, while being competitive with the Ide dec-hi approach in the feedback runs for the CACM collection. We also demonstrate the use of our user model to dynamically create a common knowledge base from the users' queries and relevant snippets using the APEX 07 data set.

Keywords: hybrid user model, information retrieval, relevance feedback, user intent, decision theory, context.

Introduction

The central idea of Computational Collective Intelligence research (CCI) is to develop theoretical frameworks and conduct empirical studies to help humans and computers to work more effectively as a group as opposed to working alone as individuals (Szuba, 2001). Thus, modeling a user is a crucial task which is central to any CCI application with regards to identifying and evaluating the knowledge of a collective as based on individual knowledge. This is also the key to addressing some of the most difficult problems encountered by the CCI community, which are the black-box difficulty and non-determinism of individual behaviors (Szuba, 2001 – pages 170-171). The main foci of the existing research in the CCI community are the use of social networks, multi-agent, ontology, and semantic web tools to determine collective intelligence (Nguyen et al., 2009). Unfortunately, the user factor does not seem to have been explored in depth in existing approaches even though a user's cognitive styles, interests, and preferences have been found to affect the way he/she interacts with computers as well as participates in the group decision making process (Gaudioso et al., 2006). In this chapter, we address this gap by presenting a hybrid approach to model a user in information seeking in which knowledge about a user is captured and used for improving his/her performance in an information seeking task. The result of this research will help address two main questions with regards to determining and evaluating collective intelligence, as pointed out in (Nguyen, 2008). First, this modeling approach captures the individual's knowledge *dynamically* and therefore would provide a computational model of non-deterministic behaviors of users while interacting with an information retrieval application. Secondly, each user model provides in-depth information about a user's goal and actions which will be used to evaluate the collective knowledge as well as analyze group performance concretely. This research also enhances the understanding of CCI by modeling individual knowledge and helps extend the use of CCI in solving any real world problems because information seeking is the key process in solving difficult real world problems.

The problem of modeling a user for information retrieval (IR) was identified as early as the late 70s and continues to be a challenge (examples include Allen, 1990; Belkin et al., 1982; Belkin & Windel, 1984; Brajnik et al., 1987; Rich, 1979; Rich, 1983; Saracevic et al., 1997; Vickery & Brooks, 1987). This addresses the lack in accounting for users and their interests within the traditional IR framework. The hypothesis is that by modeling a user's needs, better retrieval of more documents relevant to an individual user can be achieved. The main challenges of employing a user model for information retrieval arise from the fundamental questions of (i) how best to model a user so that it helps an IR system adapt to the user most effectively and efficiently; and, (ii) how best to justify the need of having a user model on top of an IR system from both the perspective of designers of an IR system and that of users of an IR system. It requires a mix of several research areas and approaches to answer these questions, such as artificial intelligence (AI) (natural language processing, knowledge representation, and machine learning), human factors (HF), human computer interaction (HCI), as well as information retrieval. The traditional framework of information retrieval supports very little user's involvement and only considers just a user's query with simple or little relevant feedback. On the other hand, the user modeling (UM) community has not taken advantage of the information retrieval techniques except for some traditional techniques such as vector space model and term frequency inverted document frequency (TFIDF) weighting.

In this chapter, we present our research towards improving a user's effectiveness in an information retrieval task by developing a *hybrid user model* that captures *a user's intent* dynamically through analyzing behavioral information from retrieved relevant documents, and by merging it with the components of an IR system in a decision theoretic framework. Capturing user intent for information seeking is an important research topic that has recently drawn much attention from both the academic and commercial communities (for examples: Baeza-Yates et al., 2006; Broder, 2002; Jansen et al., 2007; Lee et al., 2005; Rose & Levinson, 2004). Yahoo!

used to maintain a web site at http://mindset.research.yahoo.com that returned the retrieved relevant documents sorted by a user's intention). In our approach, the term *hybrid* refers to the methodology of combining the attributes describing a user's intent with the attributes describing an information retrieval system into a decision theoretic framework. The effectiveness typically refers to the number of relevant documents related to a user's problem and the relevant documents refer to the commonality order in which these documents were returned. The behavioral information from retrieved of content shared by these documents. Lastly, the components of an IR system include query, similarity measure, collection, relevancy threshold, and indexing scheme.

The novelty of our hybrid user modeling approach is that it bridges the gap between two worlds: the world of the target IR application and the world of the users of an IR application by integrating techniques from both worlds. The goal is that by using the unified decision theoretic framework, we can maximize the effectiveness of a user's retrieval task with regards to his/her current searching goals. This hybrid user model truly compensates for the lack of information about a user in an IR system by capturing user intent and by adapting the IR system to the user. It also compensates for the lack of information about an IR system within current user modeling research by using the insights from the collections to predict the effectiveness of the proposed adaptations. This is a new approach for both the IR and UM communities. This framework contributes to the IR community by using decision theory to evaluate the effectiveness of the retrieval task. This also provides the fundamental framework in modeling individual context, interests and preferences to make the job of capturing the common interests and effective cognitive styles for a group of users feasible. Additionally, our hybrid user model will provide necessary information to improve the process of determining collective intelligence based on individual knowledge and evaluating collective intelligence, which are the two challenge problems of the CCI community (Nguyen, 2008). This modeling technique also addresses the black-box and non-deterministic behaviors challenges in modeling and measuring collective intelligence (as pointed out in (Szuba, 2001)) by providing fine-grained, probabilistic representation of a user's intent in information seeking.

In what follows, we present our vision and efforts in bridging these two different areas that are crucial for building a successful information retrieval system. We also bring together some of our past results on developing and evaluating a hybrid user model for information retrieval (Nguyen, 2005; Nguyen et al., 2006) as evidence of how to apply and assess this framework. We comprehensively evaluate our hybrid user model and compare it with the best traditional approach for relevance feedback in the IR community - Ide dec-hi using term frequency inverted document frequency weighting on selected collections from the IR community such as CRANFIELD, MEDLINE, and CACM. The results show that with the hybrid user model, we retrieve more relevant documents in the initial run compared to the Ide dec-hi approach. Our hybrid user model also performs better with the MEDLINE collection compared to our user modeling approach using only a user's intent (Santos & Nguyen, 2009). We also demonstrate the use of our user model approach to dynamically create a common knowledge base from the users' queries and relevant snippets for eight analysts using the APEX 07 dataset.

The remainder of this chapter is organized as follows: We begin by reviewing key related work with regards to constructing a user model for improving retrieval performance, using decision theory for information retrieval and using user model techniques for collective intelligence. Next, some background and the description of our hybrid approach are provided. Then we present the description of the evaluation testbeds, procedures, and evaluation results, which are followed by the proposed framework of using hybrid user models for determining collective knowledge and some experimental results. Finally, we present our conclusions and future work.

Related work

The novelty of our approach is to construct a model that integrates information about a user and about an IR system in a decision theoretic framework. Therefore, in this section, we present the related methodologies for developing a user model from both the IR and UM communities, existing work on using decision theory for information retrieval, and some related work on user models used in the CCI community.

Methodologies for building a user model for information retrieval

The current approaches to building user models for IR are classified into three main groups (Saracevic et al., 1997): *system-centered*, *human-centered*, and *connections* (the latter of which we will refer to in this chapter as *hybrid* approaches).

The methods belonging to the system-centered group focus on using IR techniques to create a user model. The IR community has tried to understand a user's information needs in order to improve a user's effectiveness in an information seeking task by a number of ways such as query specification and relevance feedback/query expansion. Query specification helps a user to describe his/her information needs via a graphical user interface (such as (Michard, 1982) and (Lynch, 1992)) while relevance feedback/query expansion interactively improves a user's query by learning from the relevant and non-relevant documents in a search (e.g., Borlund, 2003; Efthimiadis, 1996; Ide, 1971; Rochio, 1971; Ruthven & Lalmas, 2003; Spink & Losee, 1996). Recently, researchers from the IR community have also applied genetic algorithms (GA) (Cecchini et al., 2008; Lopér-Pujalte et al., 2003) and support vector machine (SVM) to relevance feedback (Drucker et al., 2002) as well as applied language modeling techniques to traditional vector space model (Xie & Raghavan, 2007). Unfortunately, the system-centered approaches still do not describe entirely or even adequately a user's behaviors in an information seeking task. Our approach is different in that we determine a user's intent in information seeking to decide which concepts and relations to add to the original query instead of adding the terms based their weights directly from relevant and non-relevant documents to a user's original query.

The human-centered approaches build a user model by exploring a user's cognitive aspects in a user's interactions with an IR system. Some traditional examples include (Belkin & Windel, 1984; Brajnik et al., 1987; Ingwersen, 1992; Vickery and Brooks, 1987). The majority of these studies focus on directly or indirectly eliciting a user's interactions, preferences, cognitive searching styles, and domain knowledge to improve interfaces as well as human performance in an IR system. Recently, the information retrieval community has begun to focus more on understanding a user's information needs, a user's behaviors in an adaptive information retrieval system (e.g. Kumaran & Allan, 2008; Zhang, 2008), and a user's IR tasks and information seeking strategies (e.g. Kim, 2009) to improve user satisfaction with an IR system. Even though the findings of these studies shed some light on the possible directions for improving a traditional IR system, there are still many open questions. One criticism often raised by designers of IR systems is that any model that contains all or even a subset of a user's cognitive dimensions is highly complex and impractical. Furthermore, one key problem that arises with the approaches in this group is that they are concerned primarily with a user's behaviors and have little to say about *why* a person might engage in one particular behavior. In order to find out *why*, we have to establish the relationships between the behaviors and the goals that a user's is trying to achieve. What differentiates our approach from the approaches in this group is that we actively establish the relationship between a user's searching behaviors and a user's goals in our hybrid model.

Even though there is very little overlap between system-centered and user-centered approaches (Saracevic et al., 1997), many researchers have attempted to bridge this gap. We refer to the techniques in this category as *hybrid* approaches. There are two main views of the hybrid approaches: (i) connecting two (or more) different system-centered techniques to construct a model, e.g., applying decision theory techniques to collaborative filtering (Nguyen & Haddawy, 1999), mixing different models that represent different task-related factors (e.g. Ducheneaut et al., 2009) or temporal related factors (Billsus & Pazzani, 2000), integrating collaborative filtering techniques with knowledge-based techniques (Zanker & Jessenitschnig, 2009) or with content-based techniques (Balabanovic & Shoham, 1997); and, (ii) connecting system-centered and user-centered techniques in order to construct a model. Our approach belongs to the latter view. This direction is very challenging because finding a common ground from the different representation schemes and evaluation criteria among different disciplines (such as HF, IR, and AI) is difficult. Some historical examples can be found in the work presented in (Logan et al., 1994) and (Saracevic, 1996). In the study described in (Logan et al., 1994), Galliers' theory of agent communications is applied on the MONSTRAT model (Belkin, 1993). The STRATIFIED model proposed by Saracevic (1996) which inspired our approach in this chapter, attempted to resolve the weaknesses of both the system-centered and human-centered approaches. In the STRATIFIED model, both the user and the system sides are viewed as several levels of *strata*. Any level of the user's strata is allowed to interact with any level of the system's strata. The STRATIFIED model is constructed based on the assumption that the interactions between the user and the target IR system help the user's information seeking tasks. Within the last decade, researchers also have explored a user's searching behaviors for constructing a user model. These studies have shown that by understanding a user's searching behaviors, we develop a more flexible IR system with personalized responses to an individual's needs (Campbell & van Rijsbergen, 1996; Chen et al., 2005; Ruthven et al., 2000; Santos & Nguyen, 2009; Spink et al., 1998;). For example, in (Campbell & van Rijsbergen, 1996; Ruthven et al., 2003), temporal factor, uncertainty, and partial assessment are combined to modify the weight of a term in a relevance feedback process. The main difference between the existing approaches which incorporate a user's searching behaviors discussed above with our approach is that they use a user's search behaviors to modify the *weight of an individual term* while ours uses the captured user intent to modify the *relationships among terms* in a query. Most recently, a book edited by Amanda Spink and Charles Cole (Spink & Cole, 2005) has provided an excellent overview and new research directions to find a central ground for user-centered and system-centered approach in information retrieval. Our work here certainly contributes to this stream of research.

Decision theory for information retrieval

Even though there is some prior work from both the IR and UM communities that make use of decision theory (e.g., Cooper and Maron, 1978; Balabanovic, 1998; Brown, 1998), a decision theoretic framework that fully integrates attributes from both user and system sides has not yet been explored. From the UM community, Balabanovic (1998) has used decision theory to elicit a user's preferences over *a set of documents*. Our approach is different from his approach in that, we use decision theory to elicit a user's preferences over *a set of attributes describing a user's current search*. Brown (1998) with his *CIaA* architecture has used decision theory to determine if a user model has done a good job in assisting a user based on a set of functions measuring a user's workload, temporal, and physical efforts for a specific task. In our approach, we use the information of the components of an IR system to predict the effectiveness of a user in a current search. From the IR community, researchers in probabilistic IR have used decision theory to determine indexing schemes (Cooper & Maron, 1978), however, no attention has been paid to users.

User Model for Computational Collective intelligence

User modeling techniques have been used in building recommender systems for a group of users in entertainment domains such as choosing interesting TV programs, movies, songs for a group of users (e.g. de Campos et al., 2009; Masthoff, 2004; Pogacnik et al., 2005; Yu et al., 2006). The user models are also created for supporting people in a social network (e.g. Carmagnola et al., 2009; Upton & Kay, 2009). The user modeling (UM) researchers also focus on developing the techniques to provide and evaluate effectiveness and efficiency of group activities (please see the special issue on User Modeling to Support Groups, Communities and Collaboration, *User Modeling and User-Adapted Interactions* journal (Gaudioso et al., 2006) for a set of representative research in this area). In recent years, researchers from the CCI community have used some user modeling techniques to determine and evaluate group intelligence. In the existing literature from both the UM and CCI communities, collective intelligence is determined by finding the *similarity* of interests among the users (e.g. Barla and Bieliková, 2009; Duong et al., 2009; McDonald & Ackerman, 2000; Schmitt et al., 2003). One problem with this approach is that people who have the same perspectives, heuristics, interpretations, and prediction models may find themselves trapped/stuck at the same points when solving difficult problems (Page, 2007). Additionally, a user’s interests do not reflect *how* the users achieve or use these common interests in their tasks. Such drawbacks may decrease effectiveness in group performance. Our modeling approach presented in this paper takes into consideration both *what* a user is interested in and *how* he/she goes about achieving these interests. This serves as a concrete *first step* to help determine collective intelligence.

Capturing a user's intent in an information seeking task

Overview

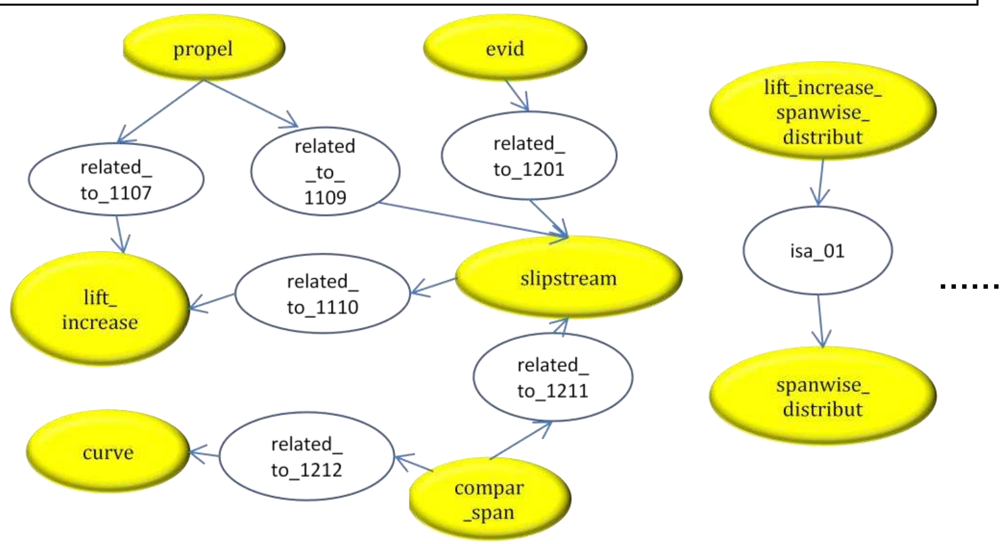
In this work, we develop a hybrid user model to improve effectiveness of an IR system. User models are needed on top of an IR system because the traditional IR framework does not utilize much input from a user except a user's query and some relevance feedback. Without a user model, it is very difficult to determine and update a user's needs. For instance, a user is searching for *“sorting algorithms”* and he possesses knowledge on *“distributed computing”* with an emphasis on “*parallel algorithms”*. He prefers to retrieve papers on specific algorithms rather than survey papers. He also prefers to retrieve as many potentially relevant documents as possible. For this user, a good IR system would display documents about parallel sorting algorithms such as “*Odd-Even sort”* or “*shear sort”* well before sequential sorting algorithms such as “*bubble sort”* or “*quick sort”*. In other words, a good IR system would pro-actively modify the original request of *“sorting algorithms”* to a request on “*parallel sorting algorithms”* which connects the user's preferences, interests, and knowledge with his current request. Additionally, in order to meet his preference to see as many potentially relevant documents as possible, the threshold for filtering irrelevant documents should be set low to allow those documents that did not contain the exact same terms but may contain synonyms or similar terms to be presented to the user.

Our goal is to improve the effectiveness of a user engaged in an information seeking task by building a user model that integrates information about an IR system and a user in a decision theoretic framework. The components of a typical IR system include *query*, *indexing scheme*, *similarity measure*, *threshold*, and *collection (*Baeza-Yates and Ribiero-Neto, 1999). Query represents a user's request. Indexing schemes contain domain knowledge represented in hierarchical relations of terms. Similarity measures are a function which determines how similar a user's query and a document from the searched collection is. Threshold is a real number which indicates how we should filter out irrelevant documents. A collection usually consists of a set of documents in a specific topic such as computer science or medicine. Usually, these components are determined when the system is developed and used. Therefore, in order to build our hybrid model, our job now is to determine information about a user and then integrate it with the components of an IR system. Our approach is to capture *user intent* in an information seeking task. We partition it into three formative components: Interests accounts for *what* a user is doing, Preferences captures *how* the user might do it, and Context infers *why* the user is doing it. The rest of this section presents the process of capturing user intent in an information seeking task (Santos et al., 2001, Santos et al., 2003b; Santos & Nguyen, 2009). In the next section, we combine the user's intent with the attributes of an IR system to create our hybrid user model.

We capture the Context, the Interests, and the Preferences aspects of a user's intent with a context network (*C*), an interest set (*I*), and a preference network (*P*). Before we describe the representation of each component, we go over how documents are represented because this representation is also used in our Context network. Each document in the target database is represented as a document graph *(DG),* which is a directed acyclic graph (DAG) containing two kinds of nodes: concept nodes and relation nodes. Concept nodes are noun phrases such as "*slipstream*" or "*comparative span*". We capture two types of relations: set-subset (denoted as "*isa*") and related to (denoted as "*related_to*"). A relation node of a DG should have concept nodes as its parent and its child. An example of a part of a *DG* generated from a document in the CRANFIELD collection is shown in Figure 1. The highlighted terms in the Figure 1(a) are concept nodes in the Figure 1(b). We developed a system to automatically extract *DG* from the text (Zhao et al., 2009). We extracted noun phrases (NPs) from text using Link Parser (Sleator & Temperley, 1993); these NPs will become concept nodes in a *DG*. The relation nodes are created by using three heuristic rules: *noun phrase heuristic, noun phrase-preposition phrase heuristic*, and *sentence heuristic* (see Santos & Nguyen, 2009 for specific details). Each query is also represented as a query graph *(QG)* which is similar in representation to the document graph.

Experimental investigation of the aerodynamics of a wing in a slipstream. An experimental study of a wing in a propeller slipstream was made in order to determine the spanwise distribution of the lift increase due to slipstream at different angles of attack of the wing and at different free stream to slipstream velocity ratios. The results were intended in part as an evaluation basis for different theoretical treatments of this problem. The comparative span loading curves, together with supporting evidence, showed that a substantial part of the lift increment produced by the slipstream was due to a /destalling/ or boundary-layer-control effect. The integrated remaining lift increment, after subtracting this destalling lift, was found to agree well with a potential flow theory. An empirical evaluation of the destalling effects was made for the specific configuration of the experiment

(a)



(b)

Figure 1: (a) An example of a document from the CRANFIELD collection. (b) A small part of the document graph generated from this document

Interest Set

The *Interest set* represents *what* a user is currently focusing on. Each element of this set describes an interest concept *a* and its associated interest level *L(a)*. Interest concept refers to the terms that the user is interested in and interest level is a real value in the range of [0,1] that represents the user's emphasis on a particular concept. We use the intersection of *DGs* of retrieved relevant documents to create and update an interest set. The algorithm of finding the intersections of retrieved relevant documents is shown in the Figure 2. Each concept in this intersection will be added to the current interest set with the interest level being the ratio of frequency of a specific concept node over the total concept nodes in the intersection and if this interest level is greater than the user-defined threshold for Interest list.

```
/* D is the set of m retrieved relevant document graphs */
Vector Intersection(D) {
        Vector J= ∅
        For i = 1 to m {
                For each node c in D_i do
                    frequency(c)=0
                    If  c is a concept node then
                          For j = 1 to m
                               If  i does not equal to j and (D_j contains c) then
                                    frequency (c) ++
                                    if (frequency(c) > threshold then J = J + c
                                                        }
        For i=1 to m do {
                For each node r in D_i do
                    frequency(r) = 0
                    For k=1 to m do
                          If   (r is a relation node and (its parent and its child are in J and
                               k does not equal to i) then frequency(r) ++
                If frequency(r) > threshold then J = J +  (r's parent -  r - r's child)
        }
        return J
}
```

Figure 2: Pseudo code of algorithm to find intersections of retrieved relevant documents

A user's interests change over time, therefore, we re-compute the interest level *L(a)* for each concept *a* in the existing interest set after each query as follows:

$$-\tag{1}$$

in which *p* as the number of retrieved relevant documents containing *a* and *q* as 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.

Context Network

The Context network captures a user's knowledge in a specific domain, which is represented in terms of concepts and relationships among these concepts. The basic structure of the Context network is similar to the representation of a *DG*. We choose the *DG* representation for representing a Context network because it satisfies two requirements: (i) it can be generated dynamically, and (ii) it represents concepts and relationships among concepts visually. In a Context network, each node also has a weight, value, and bias. These attributes are used in our re-ranking algorithm to infer the current interests. 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 weight equal to 0.8 in this implementation. The value of a node represents its importance to a user and is a 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 are used by a spreading activation propagation algorithm. The basic principle of this algorithm is that a node located far from an interest concept, which has been indicated as being interested by the user, will be of less interest to the user. The spreading activation propagation algorithm (Santos & Nguyen, 2009) is shown in Figure 3.

Note that in this algorithm, we treat the nodes with one parent differently from those with multiple parents. The intuition is that a single parent only will have stronger influence on its children while the influence from multiple parents needs to be aggregated to avoid bias from a specific parent.

A Context network is dynamically constructed by finding the intersection of all document graphs representing retrieved relevant documents. The algorithm to find the intersection of the retrieved relevant document is shown earlier in the previous subsection. We use the set of common sub-graphs of the retrieved relevant documents in the intersection. If a sub-graph is not currently in the Context network, it is added to the Context network accordingly. If the addition results in a loop, it will be skipped. We do not consider loops because we want to maintain the rationality of the relationships among concepts and also ensure that our spreading activation inference algorithm on a Context network terminates properly. This does not cause any problems because of the semantics of the relationship that we capture. Specifically, a loop cannot occur for "*isa"* relationships. If a loop occurs for "*related_to*" relationships, this is already covered by the transitive and reflexive attributes of this type of relationship so the extra link is unnecessary.

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. An example of a Context network of a user model from one of the experiments conducted in this chapter is shown in Figure 4.

```
Vector SpreadingActivation(I, Q, C) {
    /*Initializing bias is executed as follows:  (i) set the bias equal to 1 for every concept
    found both in the current context network and in the current query graph; and (ii) set bias
    equal to the interest level for every interest concept found both in the current context
    network and in the current interest set. */
    initBias();

    /* Then we compute and sort all the nodes in the concept network C by its depth.
       d(a) = 0 if the node has no parents
       d(a) = max(d(p(a))) + 1 with p(a) is a parent of a.
    */
    Vector  nodesInContextNetwork = sortBasedOnDepth(C);
    For each node a in C do {
        If (a.getNumberOfParents() == 0) then {
            sum = 0
            if (bias > sum) then value = (sum + bias) / 2
            else value =0
        }

        If (a.getNumberOfParents() == 1) then {

            sum = value(p(a))* weight(p(a))
            if (bias > sum) then value = (sum + bias) / 2
            else value = sum
        }
        If (a.getNumberOfParents() > 1) then {
            sum = 1 / (1 + e^(-(Σ value(p_i(a))*weight(p_i(a)))/n))

            /* 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 */
            if (bias> sum) then value = (sum + bias) / 2
            else value = sum

        }
    }
    Vector newInterests =
    Vector conceptList = sortConceptNodesByValues(C);
    For each c in conceptList do {
        If (c.getValue() >= short term interest threshold) then
            newIntersts=newInterests + c
    }
    Return newIntersts
}
```

Formulas in the listing:

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

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

Figure 3: Pseudo code of our spreading activation algorithm

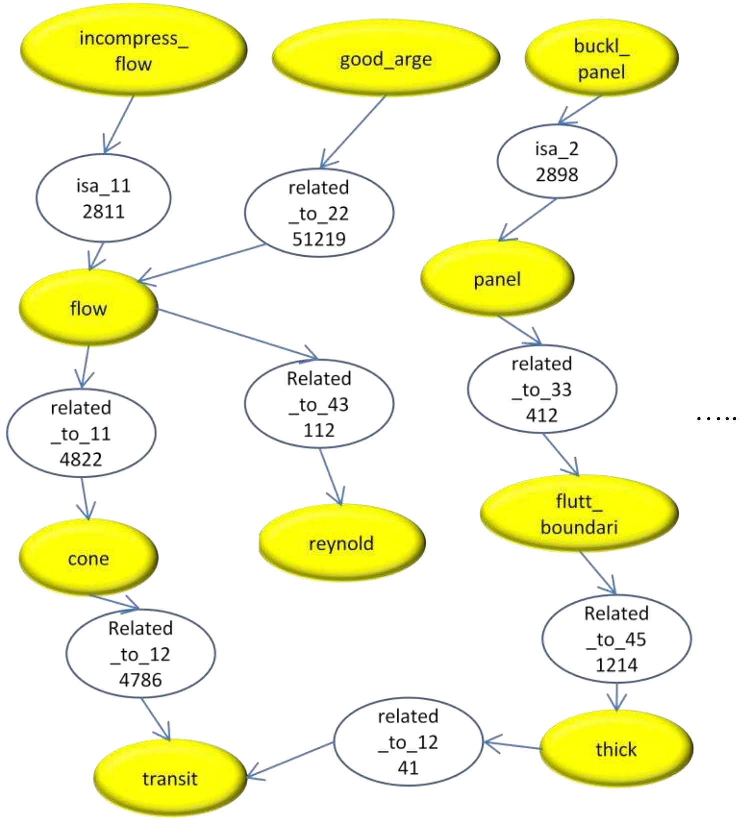


Figure 4: A part of a user's context network

Preference Network

The *Preference network* represents *how* a query is formed (Santos & Nguyen, 2009). We chose Bayesian networks (Jensen, 1996) to represent a preference network because they offer visual expressiveness, and ability to model uncertainty. A user shows his/her preferences in the ways to modify a query by choosing from a class of tools. We currently choose two tools to implement in our Preference network which are filters and expanders. A filter is a tool that searches for documents that narrows the search topics semantically and an expander that searches for documents that broaden the search topics semantically. We chose these two tools because they are similar to two typical methods of refining a search query in information seeking: specification and generalization (Lau & Horvitz, 1999).

There are three kinds of nodes in a preference network: pre-condition node P_c, goal node G, and action node A, as shown in Figure 5. A P_c represents the requirements of a tool, such as the query and/or the concepts contained in the current interest relevancy set. If a P_c represents a concept from a user's interest set, its prior probability will be set as its interest level. If a P_c represents a query, its prior probability will be set as its frequency. A goal G represents a tool, which is again a filter or an expander in the current design. The conditional probability table of each goal node is similar to the truth table of logical AND. Action node A represents an action associated with each goal node. 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.

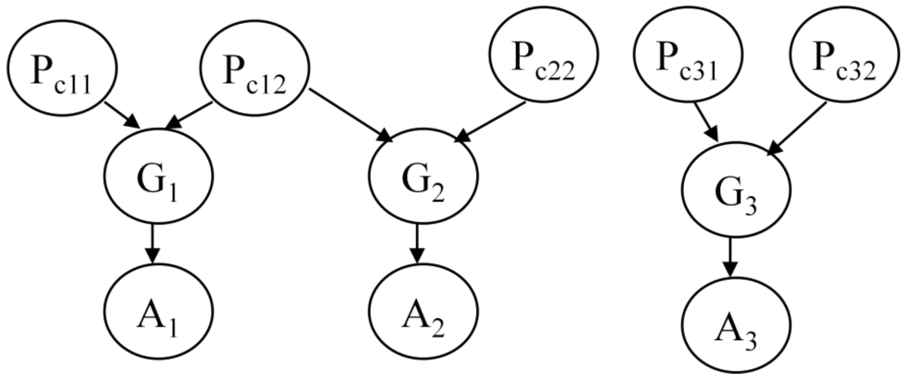


Figure 5: Conceptual structure of a preference network

To manage the size of the conditional probability table of each goal node when the interest set grows, we apply a divorcing technique (Pearl, 1988) through noisy-AND nodes. An example of a preference network is shown in Figure 6. There are five pre-condition nodes in this network in which three pre-condition nodes represent the user's interests (labeled as "*condit", "shock"*, and *"equilibrium"*) and two nodes represent the user's queries (labeled as "*query_01"* and "*query_02"*). The highlighted nodes in this figure are the ones set as evidence. The interest concept "*equilibrium*" has interest level as 0.55 and therefore its prior probability is set as (P(*equilibrium* = true)=0.55 and P(*equilibrium* = false) = 0.45) as shown in Figure 6. Similarly, the prior probability of the pre-condition node query_02 is set as P(*query_02*=true) = 0.8 and P(*query_02*=false)=0.2. There are two goal nodes and two actions in this preference network. Figure 6 also shows the conditional probability tables associated with the goal nodes *filter_01* and *expander_01*, which are similar to the logical AND truth table. Each entry in a conditional probability table of a preference network shows the conditional probability of each outcome of the goal node given outcomes of all pre-condition nodes (e.g. P(*filter_01=true|query_01=true*, *condit=true*, *shock=true*) = 1). A conditional probability table associated with an action node shows the probability of each outcome of the action given the outcome of each goal (e.g. P(*proactive_query_01=true|filter_01=true*) = 1).

We use a user's query and retrieved relevant documents to create and update 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 corresponding 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. A new tool is chosen based on the probability that a new network with this tool will improve the user's effectiveness. Specifically, we determine how frequent this tool has helped in the previous retrieval process. Currently, if the total number of retrieved relevant documents exceeds a user-defined threshold, the tool used for the query modification is considered helpful.

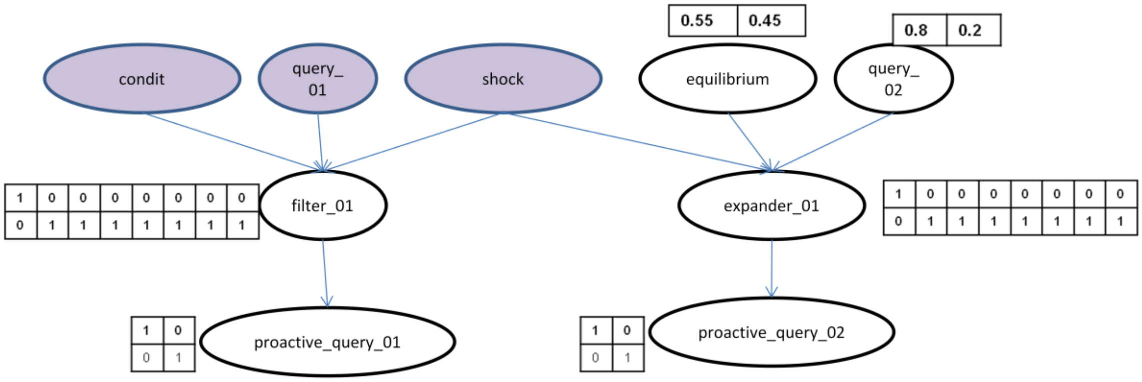


Figure 6: Example of a preference network

Hybrid User Model

Overview

We combine the user intent with the elements of an IR application in a decision theoretic framework to construct a hybrid model to improve a user's effectiveness in a search. Our solution is to convert this problem into a multi-attribute decision problem and use multi-attribute utility theory (Keeyney & Raiffa, 1976) in which a set of attributes is constructed by combining the set of attributes describing a user's intent and the set of attributes describing an IR system. In a multi-attribute utility model, the decision is made based on the evaluation of *outcomes* that are generated by the actions performed by a user. In this problem, the outcome space represents the set of *all* possible combinations of a user's intent and components of an IR system. Each outcome represents a specific combination of a specific user intent and specific collection, indexing scheme, query, similarity measure, and threshold. There are two reasons for using this multi-attribute utility model. First, the estimation of a user's effectiveness in a search with respect to the searching goal is a problem of preference elicitation. It represents a user's preferences over a space of possible sets of values describing a user and describing an IR system. Second, the framework of a multi-attribute decision problem allows us to use the elicitation techniques from the decision theory community to decide which combination will be likely to produce the most effective search. We use the literature on predicting query performance in IR (He & Ounis, 2004) and computing dissemination thresholds in information filtering (IF) (Boughanem & Tmar, 2002) in our elicitation technique.

In this hybrid user model, eight attributes are initially considered from the set of attributes describing user intent and the set of attributes describing an IR system. They are: *I* (a user's Interest), *P* (a user's Preferences), *C* (a user's Context), *In* (Indexing scheme), *S* (similarity measure), *T* (threshold), *D* (Document collection) and *Q* (a user's query). From this list of eight attributes, we know that *I*, *P* and *C* have been used effectively to modify a user's query *Q* (Santos et *al.*, 2003a, Santos & Nguyen, 2009). Therefore, the attribute *Q* can subsume the attributes *I*, *P* and *C*. In the traditional IR framework, the indexing scheme *In*, document collection *D*, and similarity measure *S* are decided when designing a system and shall remain unchanged during a search process. These three attributes do not participate in the decision making process.

After reducing the number of attributes to core attributes, we focus on only two attributes *Q* and *T*. Note that this is not a simple integration between query and threshold attributes because we model the non-deterministic behavior of a user in information seeking through a user's interests, preferences, and context and use this information to modify a user's query. Therefore, by using these two attributes, we reflect well both a user and a typical IR system and reduce the complexity of the problem. We then evaluate each outcome through a real value function. We also make an assumption that these two attributes are preferentially independent (Keeyney & Raiffa, 1976). Thus, this value function representing a user's preferences over these two attributes can be defined as follows:

(2)

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

In this hybrid user model, we do not work directly with the value functions because it is very difficult to elicit the coefficients λ_i. Instead, we determine the partial value function which consists of two sub-value functions: one over query and one over threshold. The partial value function implies that an outcome x_1 with the value (x_{11}, x_{12}) is preferred to an outcome x_2 with value (x_{21}, x_{22}) if and only if

x_{1i} x_{2i} for all $i=1,2$, and

$x_{1i} > x_{2i}$ for some i.

For each sub-value function, each attribute needs to be evaluated with respect to a user's effectiveness in achieving a searching goal at a given time. We assume that a user's searching goal at any given time is to retrieve many relevant documents *earlier* in the information seeking process. Therefore, we choose the average precision at three point fixed recalls as the effectiveness function because it measures both the percentage of retrieved relevant documents and the speed of retrieving these documents. This function is computed by calculating the precision values at various recall points. For instance, we calculate the average precision (*ap*) at three point fixed recalls of 0.25, 0.5, and 0.75 for a query Q by:

(3)

where P_{recall} is computed as follows:

$$P_{recall} = \qquad (4)$$

where m_{recall}=recall*N with N is the total relevant documents for this query Q, and M is the number of the retrieved documents when m_{recall} relevant documents are retrieved.

Sub-Value Function over Query

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

Recall that each query in this approach is represented by a query graph which contains relation and concept nodes (Santos et *al.*, 2003a; Santos & Nguyen, 2009). Therefore, we created the sub-value functions for the concept nodes and for the relations. A sub-value function for the concept nodes is computed as follows:

$$V_c(Q) = \qquad (5)$$

in which

$$– \qquad (6)$$

with *n* as the number of concepts in *Q*

$$(Q) = \qquad (7)$$

and,

$$idf_c(c) = \qquad (8)$$

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

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

$$V_r(Q) = \qquad (9)$$

in which

$$– \qquad (10)$$

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

$$(Q) = \qquad (11)$$

and,

$$idf_r(r) = \qquad (12)$$

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

Sub-Value Function for Threshold

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

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

$$T_0 = p * N_0 \qquad (13)$$

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

$$p = – \qquad (14)$$

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

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

$$T_{(t+1)} = T_t + \qquad (15)$$

where λ=1300 and φ=500 and R_t is the total number of relevant documents at time *t*, and d_{last} is the similarity of the last retrieved document in the previous retrieval. The values of these λ and φ constants are obtained from experimental results in (Boughanem & Tmar, 2002). The logic for this approach is that if the number of retrieved relevant documents is small, and the difference between the similarity of the last returned documents and the threshold is big, then we need to decrease the threshold considerably in order to retrieve more relevant documents. Otherwise, we may not need to decrease the threshold.

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

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

(16)

Complexity of Hybrid User Model

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

Implementation

We embed and use this hybrid user model in an IR system as follows:

- A user logs into an IR system. If the user is new, then he/she is asked for his/her preferred percentage of documents needed to be returned *p*.
- The user issues a query *Q*. The user's query is modified using the information contained in the Interest, Preference and Context (the pseudo code is shown in Figure 7). Assuming that there are *m* goals fired in the Preference network, each goal generates a query, so we have the query sets $\{Q_1, Q_2,\dots, Q_m\}$.
- Use the sub-value function to evaluate each Q_i. Choose the query with the highest sub-value function evaluation. Determine T_0 for initial threshold.
- Send the query with the highest value evaluated by the sub-value function to the search module, perform the search, filter out the documents based on the value of the threshold, and display the results to the user.

- After reviewing papers, we update the sub-value function *V(T)*. If a new query is issued, re-compute the threshold depending on the number of documents seen in the previous step.

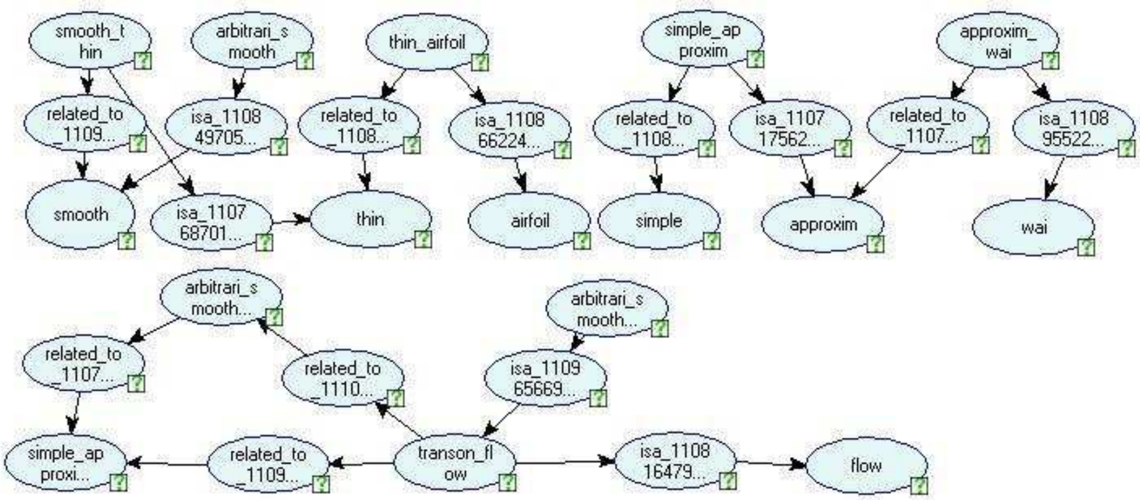
ModifyQuery (I,P,C,Q) {

- Using the spreading activation algorithm described earlier to reason about the new set of interest *I'*.
- Set as evidence all concepts of the interest set *I'* found in *P*.
- Finding a pre-condition node representing a query in *P* which has associated *QG* that completely or partially matches against *q*. Set it as evidence if found.
- G = Performing belief updating on the preference network P. Choose top *m* goal nodes from preference network with highest marginal probability values.
- For every goal node in *G* do
  - If the query has been asked before and the user has used this goal, replace the original query subgraph with the graph associated with the action node of this goal.
  - If the query has not been asked before and the goal node represent a filter:
    For every concept node q_i in 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 c_{ii} in the context network such that c_{ii} is an ancestor of cq_i. If such c_{ii} 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 c_{ii} in the context network such that c_{ii} is a progeny of cq_i. If such c_{ii} and cq_i are found, we add the paths from context network between these two nodes to the modified query graph.
  - Add this modified query graph Q_i to the query set

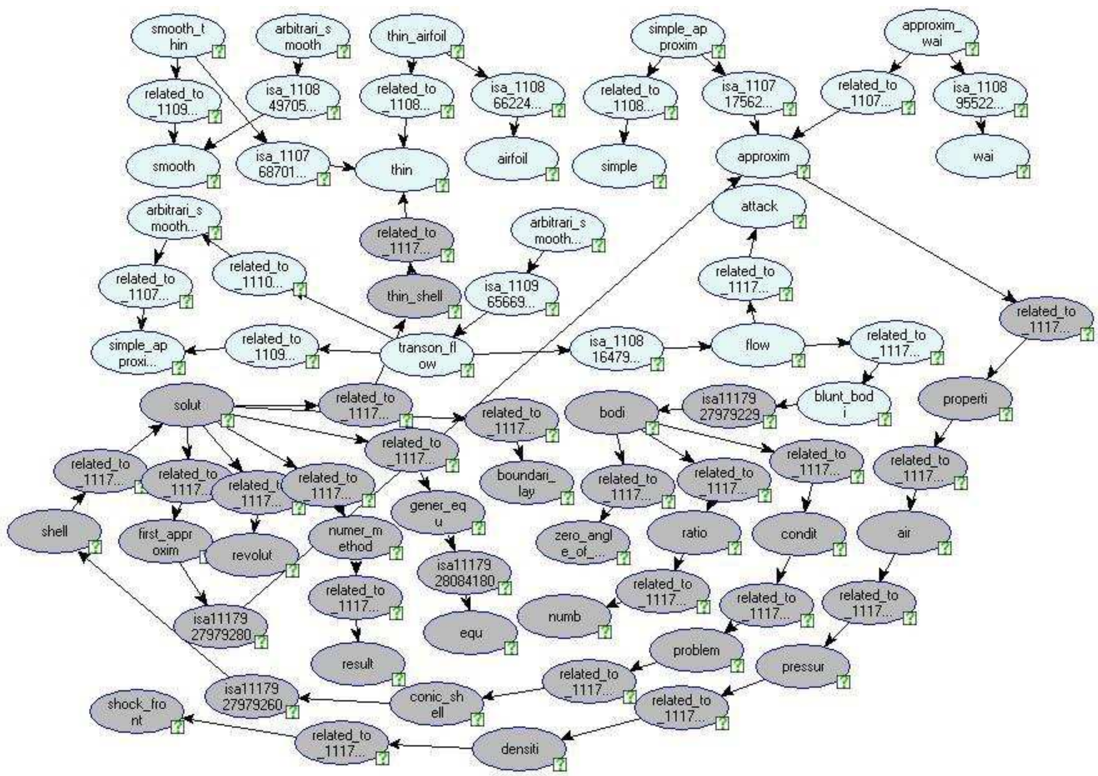
}

Figure 7: Pseudo code for modifying a user's query using captured intent.

An example of an original query and a proactive query is shown in Figure 8 (a) and (b) respectively. The shaded nodes in Figure 8(b) are the nodes added as a result of the algorithm in Figure 7.



(a)



(b)

Figure 8: (a) an example of an original query (b) the corresponding proactive query.

Evaluation

Objectives

The objectives of this evaluation are as follows:

- To assess whether the hybrid user model improves user effectiveness in an information seeking task. The intuition behind this objective is that we want to verify if our user model helps users retrieve *more* relevant document *earlier* in an information seeking process.

- To compare our hybrid user model with the existing approaches from the IR community by using collections, metrics and procedures from the IR community. We set this objective to see where our approach stands and what we can do to make it work better.
- To assess the effect of having a long-term user model with an IR system. User models offer a way to re-use knowledge in the past to speed up and enhance the information seeking task at present. With this objective, we hope to understand more about the effectiveness of this reusability in combination with the change of a user's current focuses.

Testbeds

We decide to use three small collections, CRANFIELD, CACM, and MEDLINE, because a common testbed for evaluating an adaptive IR system still does not exist (Voorhees, 2008). Also, these three collections have been extensively studied in our previous work (Nguyen et al., 2004a; Nguyen et al., 2004b; Santos & Nguyen, 2009). Even though these collections are not made for evaluating user modeling techniques, their sizes are appropriate and they have been used in the IR community to evaluate the effectiveness of relevance feedback techniques (Drucker et al., 2002; Lopéz-Pujalte, 2003; Salton & Buckley, 1990). The CRANFIELD collection contains 1400 documents and 225 queries on aerodynamics; CACM contains 3204 documents and 64 queries in computer science and engineering (CSE); while MEDLINE contains 1033 documents and 30 queries in the medical domain (Salton & Buckley, 1990). We use the complete set of queries from these collections in our evaluation.

Vector space model and Ide dec-hi

We compare this user modeling approach against the vector space model using term frequency inverted document frequency (TFIDF) weighting scheme and the Ide dec-hi technique for relevance feedback (Salton & Buckley, 1990). TFIDF and the Ide dec-hi techniques are very well-documented (Frake & Baeza-Yates, 1992; Lopéz-Pujalte et al., 2003; Salton & Buckley, 1990) and thus make it easier and more reliable for re-implementation. Secondly, the Ide dec-hi approach is still widely considered to be the best traditional approach in the IR community (Frake and Baeza-Yates, 1992; Lopéz-Pujalte et al., 2003; Salton & Buckley, 1990). It offers an opportunity to see where we stand with other approaches as well.

The main idea of the Ide dec-hi is to merge the relevant document vectors into the original query vector. This technique automatically re-weights the original weight for each term in the query vector by adding its corresponding weights from the 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:

$$Q_{new} = Q_{old} + \qquad - D_j \tag{17}$$

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.

Procedures

We initially followed the traditional procedure for evaluating any relevance feedback technique as described in (Salton & Buckley, 1990). However, this procedure did not provide a way to assess the new features of our hybrid model. Thus, we employ a new evaluation procedure to assess the use of knowledge learned over time.

Traditional procedure

We first apply the traditional procedure used in (Salton & Buckley, 1990) for both Ide dec-hi/TFIDF and the IR application enhanced by our hybrid model. We issue each query in the testbed. We then identify the relevant and irrelevant documents from the first 15 returned documents, and use them to modify the query proactively. For the Ide dec-hi/TFIDF, the weight of each word in the original query is re-computed using its weights in relevant documents and the first irrelevant document. The words with the highest weights from relevant documents are also added to the original query. For our user modeling approach, we start with an empty user model and add the concept and relation nodes to the original query graph based on the procedure described in previous sections. The structure of a query graph is similar to the structure of a document graph and we construct it from a user's query. We choose to use the sub-value function $V_c(Q)=\sigma_{idf\text{-}c}(Q)$ over concept nodes in a query as a sub-value function for the query because it is simple and easy to implement. We then run each system again with the modified query. We call the first run, the *initial run* and the second run, the *feedback run*. For each query, we compute average precision at three point fixed recalls.

Procedure to assess long-term effect

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

Results and Discussion

Results of traditional procedure

The average precision at three point fixed recall of the initial run and feedback run using original collection of the experiments in standard procedure for CRANFIELD, CACM and MEDLINE is reported in Figure 9. Also in this figure, we report the results for TFIDF/Ide dec-hi approach. In the traditional procedure, it shows that using CRANFIELD collection, we achieve better results in both initial run and feedback run compared to TFIDF/Ide dec-hi approach. For the MEDLINE collection, we clearly achieve better results in the initial run compared to TFIDF approach. Lastly, we achieve competitive performance using CACM collections compared to Ide dec-hi with TFIDF.

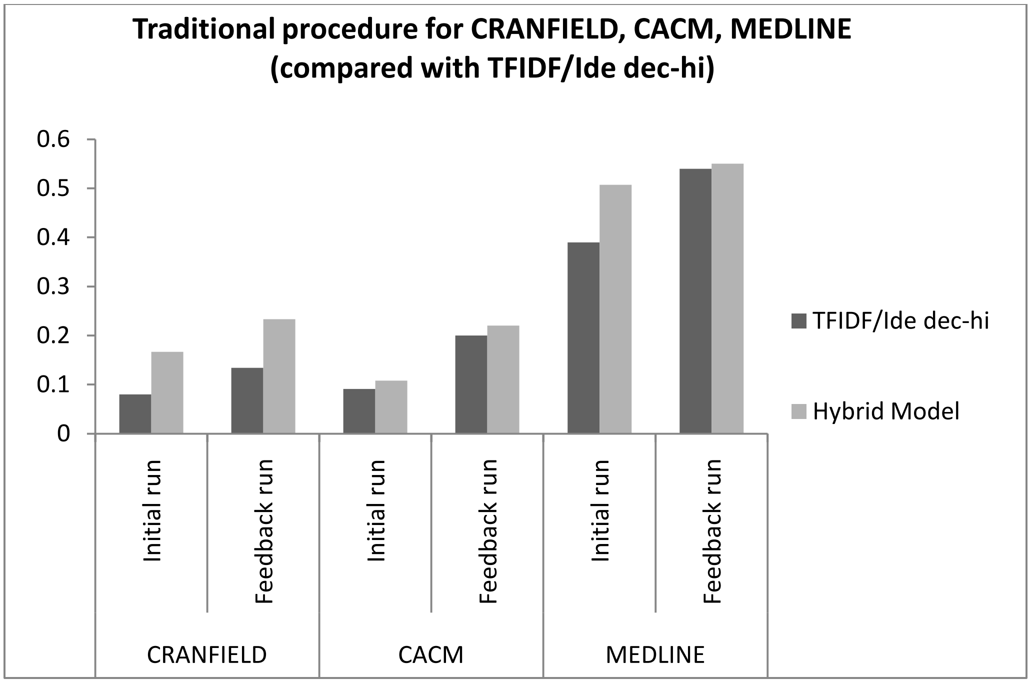


Figure 9: Result for traditional procedure while comparing with TFIDF/Ide dec-hi.

Compared to our user model that contains only a user's intent on the entire CACM and MEDLINE collections (that model is referred to as *IPC* model in (Santos & Nguyen, 2009)), as shown in Figure 10, we achieve only competitive results in both runs for the CACM collection while we are clearly better in the initial run and competitive for feedback run for the MEDLINE collection. In the earlier evaluation done with the *IPC* model, we only selected a small subset of queries from the CRANFIELD collection while we used the entire CRANFIELD collection in this evaluation for the hybrid model. Therefore, no comparisons were made with CRANFIELD.

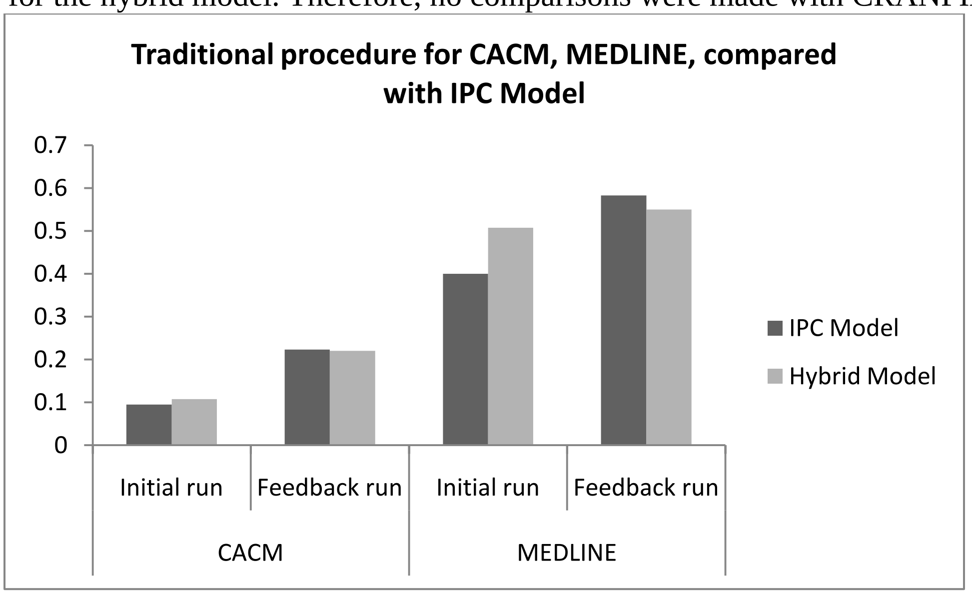


Figure 10: Result for traditional procedure while comparing with IPC model

Results of new procedure to assess long-term effect

The results of our procedure to assess the long-term effects of our hybrid approach are shown in Figure 11. It shows that by using our hybrid model, the precision of the feedback runs is always higher than those of the initial runs. In the MEDLINE collection, for example, our initial run using knowledge of learned queries is even better than the feedback run of Ide dec-hi/TFIDF. That means the relevant documents are retrieved earlier in the retrieval process than the other approach. For the CRANFIELD collection, we outperform the TFIDF/ Ide dec-hi approach in both initial and feedback runs. For the CACM collection, with the new procedure, we maintain the trend of retrieving more relevant documents in the initial run compared to TFIDF approach (0.144 vs 0.091). If we compare these results with our previous results using the model with only information about a user's intent in (Nguyen et *al.*, 2004a, Santos & Nguyen, 2009), we achieve only competitive results in both runs for the CACM collection while we are clearly better in the initial run and competitive for feedback run for the MEDLINE collection (as shown in Figure 12).

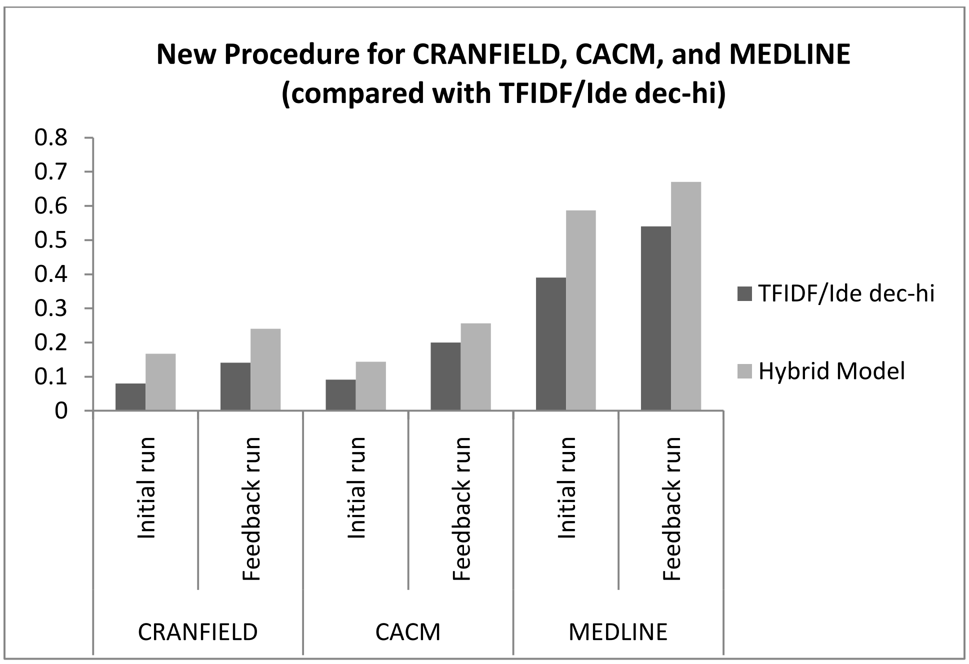


Figure 11: Results for the procedure to access long-term effect compared with vector space model/Ide dec-hi.

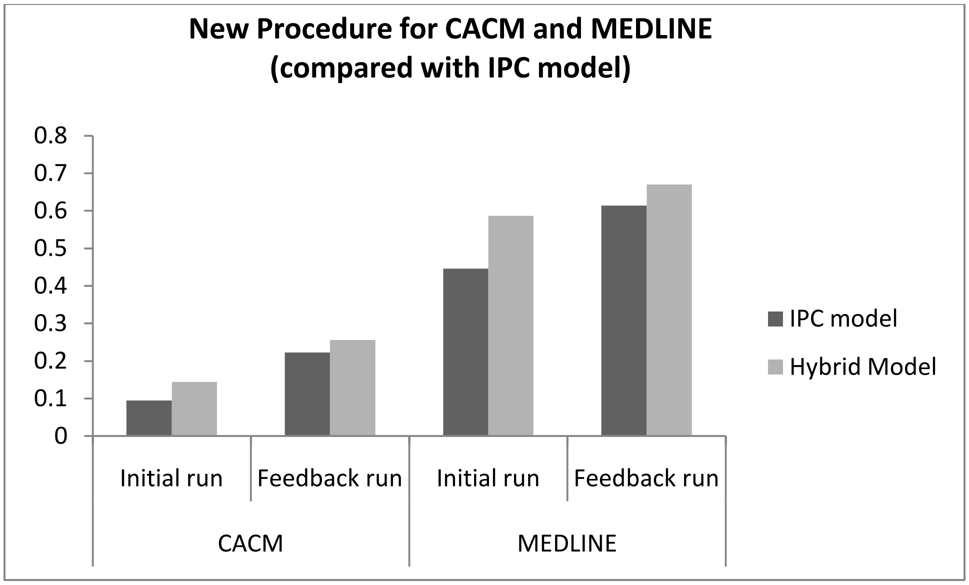


Figure 12: Results for the procedure to access long-term effect compared with the IPC model

Discussion

In summary, with regards to our objectives, we have found the following from this evaluation:

- Effectiveness of the model: Our hybrid user models help retrieve more relevant documents earlier in the initial run compared to both vector space model with TFIDF and *IPC* model.
- Usefulness of having a long term model: In this hybrid model, the long-term and short-term interests of a user in information seeking are combined and balanced. The long-tem interests here refer to knowledge learned from a query or a group of queries while short-term interests refer to knowledge learned from a current query. We clearly perform better than the traditional vector space model with TFIDF/Ide dec-hi on the CRANFIELD and MEDLINE collection and we are competitive on the CACM collection.

As we may have noticed, from the results in both the experiments with traditional procedure and new procedure, our hybrid user model performs better in the initial run but only performs competitively in the feedback run compared to the vector space model using TFIDF weighting scheme and Ide dec-hi approach. The main reason is that our hybrid user model retrieved more relevant documents in the top 15 in the initial run, as shown in Figure 13. Therefore, as a result, there are less relevant documents left to be retrieved in the feedback run.

In comparison with the *IPC* model, we achieve only competitive results for the CACM but clearly perform better for the MEDLINE. The main reason for this behavior of the hybrid model depends on the distribution of the value function for queries, the distribution of the standard deviation of *idf* for retrieved relevant documents and all relevant documents, and the tools that are chosen to modify each query in the testbed. Further studies on this will be conducted in our future work.

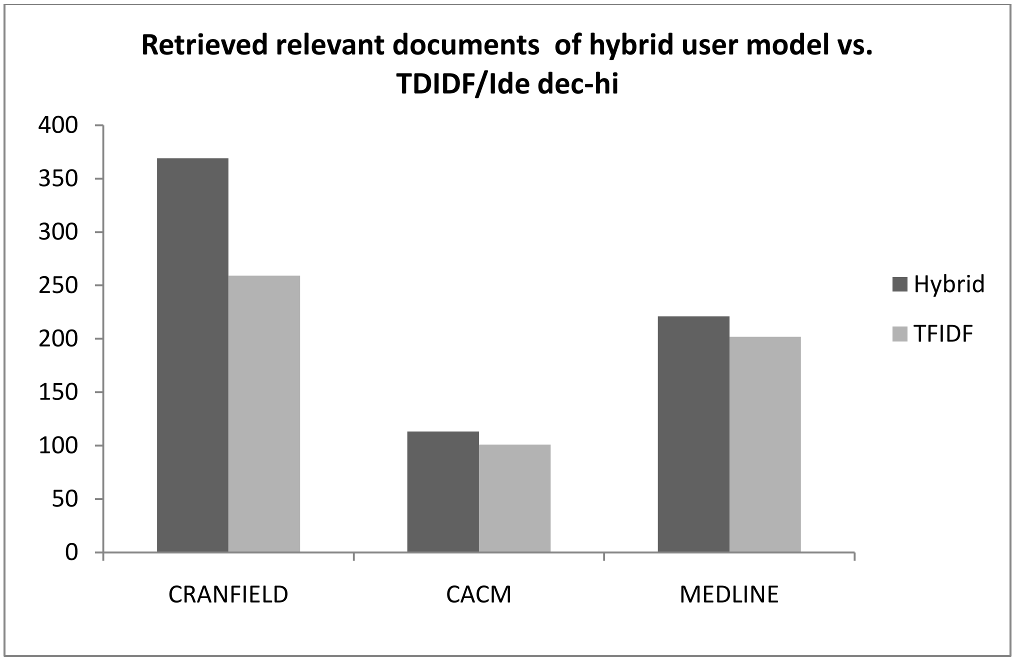


Figure 13: Retrieved relevant documents in the top 15

Hybrid User Model and Collective Intelligence

In this section, we propose a framework in which our hybrid user model can be used to address two fundamental questions of the CCI community (Nguyen et al., 2009) which are (i) how to determine the collective knowledge based on the individuals' knowledge, and (ii) how to evaluate the collective knowledge. Our proposal will demonstrate how to use our user modeling technique in order to develop collective knowledge and collective strategies in the CCI community.

Determining the collective knowledge from individual knowledge

Our hybrid user models can be used to dynamically create the common knowledge bases that contain general domain information as well as provide effective ways of learning a specific domain. Figure 14 describes the pseudo code for creating a general knowledge base that contains the concepts and relationships of concepts from a set of Context networks. The novelty of this approach as compared to static knowledge bases such as Wordnet (Miller, 1995) is that the collective domain knowledge is generated dynamically from a user's interactions in an information seeking process. This approach also differs from the approaches that use traditional vector space model with TFIDF to general domain ontology (such as the work in Duong et al., 2009) in that it provides the relationships between concepts including the concepts that do not occur in the same document as opposed to a list of independent concepts.

```
CommonKnowledgeBase(HybridUserModel u[]) {
        knowledgebase = null;
        For each hybrid user model u_i do {
                Retrieve the context network: c_i = u_i. getContextNetwork();
                Node n_i = c_i. getFirstNode();
                while ( n_i !=null) {
                    if (knowledgebase does not contain n_i and n_i is a concept node) {
                            Add n_i to knowledgebase;
                            Find all of its grand children to see if they are included in
                            the knowledgebase. If they are, add the links
                            Find all of its grant parents to see if they are included in the
                            knowledgebase. If they are, add the links.
                        }
                        n_i = c_i.getNextNode(n_i);
                } // end of while
        } //end of for
        Return knowledgebase;
}
```

Figure 14: Pseudo code of creating common knowledge base from a set of hybrid model.

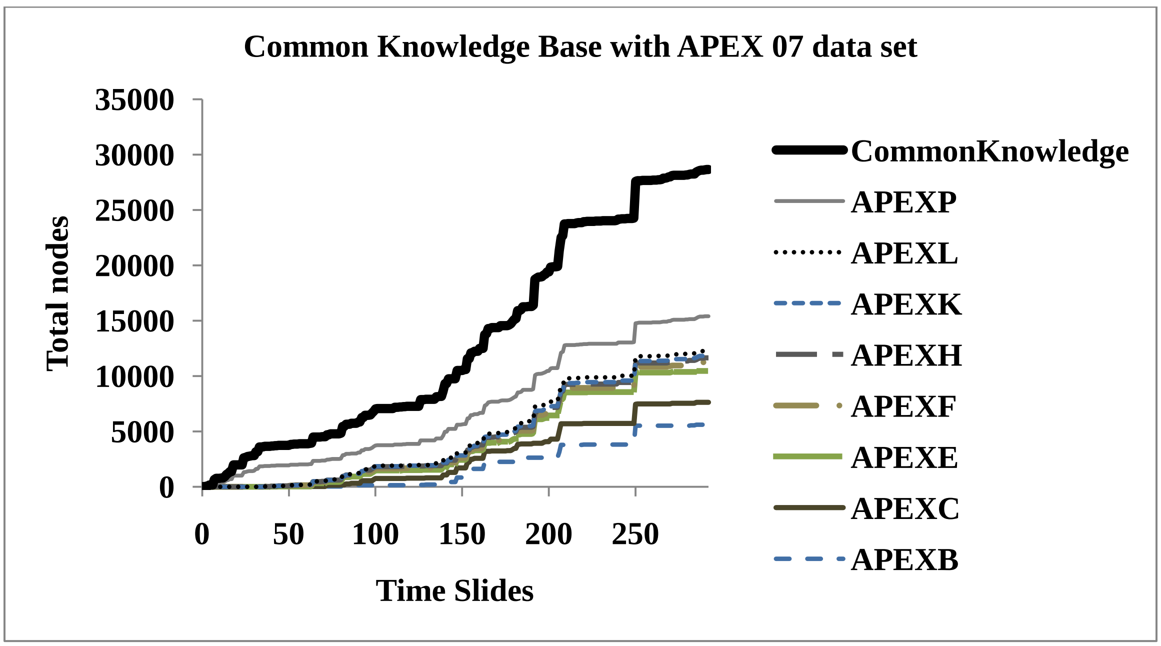


Figure 15: Common Knowledge Base is created using the APEX 07 data set.

We have created a common knowledge base from a set of queries and relevant snippets that were issued by eight intelligent analysts in the APEX 07 data set. This data set was created by the National Institute of Standards and Technology (NIST) to simulate an analytical task in the intelligence community. This collection included eight analysts (namely APEXB, APEXC, APEXE, APEXF, APEXH, APEXK, APEXL, and APEXP), their recorded actions over time, and their final reports. In the APEX data set, there are eight types of actions: "*Start application*", "*Search*", "*Retain*" (happens when an analyst bookmarks, prints, saves a document, or cuts and pastes information from a document to his/her report), "*Access*" (happens when an analyst views a document), "*Make Hypothesis*", "*Associate Evidence*" (happens when analyst links a document or a snippet to a hypothesis), "*Assess*" (happens when analyst assesses how relevant a document or snippet is to a hypothesis), and "*Discard*" (happens when a user discards evidence). In this experiment, we create a user model for each analyst using only his Retain and Search actions. Figure 15 shows the common knowledge base is created by using the algorithm shown in Figure 14 for these eight analysts from 12/07/2007(15:42:44) to 12/11/2007 (14:03:58).

We can also use our hybrid user models to develop a collective strategy by constructing a library of the most effective queries. This is done by combining the set of queries from users and ranking them by the value of the sub-value function over each query. The queries used in the IR application enhanced by this hybrid user model are the ones with highest value functions for each user. Figure 16 shows a pseudo code of the algorithm to create a library of queries for a group of users. The query library can be used to improve the retrieval effectiveness of the collaborative information retrieval activities in which concepts from the queries or queries themselves can be recommended to the users of a collaborative group to improve the number of relevant documents retrieved in the information seeking process. For our experiments with CRANFIELD, CACM and MEDLINE collections in this paper, we currently maintain the library with 225, 64 and 30 proactive queries for these three collections, respectively.

Evaluation

One of the biggest challenges in CCI research is the evaluation of the collective knowledge in order to compare group performance against individual performance and to validate the hypothesis that the collective knowledge helps improve the effectiveness of group performance. Unfortunately, this evaluation problem has yet to be explored in depth in the CCI community. Our hybrid user modeling technique can help address this problem. First, in order to compare group performance against individual performance, we can use the common knowledge base and common query library created above for a group's collaborative activities. We will then run an IR application with these common knowledge bases and with each individual model. Next, we compare the retrieval performance for the same lists of queries.

Second, we can take advantages of the existing testbeds in the information retrieval community as well as information seeking, information analytic domains to create testbeds for collective intelligence applications. Small collections such as CACM, MEDLINE, CRANFIELD can be used as well as other collections such as APEX 07 collection for analytic domain and CNS collection for information seeking domain can also be used (Santos et al., 2005). Both of these directions are currently being pursued by our group.

```
CommonQueryLibrary (HybridUserModel u[]) {
    Library = null;
    For each user model u_i do {
        Retrieve a user's current set of queries: q_i = u_i.getQuerySet();
        Sort this set by values of value functions over query
        For each query in this set q_ij do {
            If (Library contains q_ij) {
                If (current values for the q_ij in the Library < values for
                this q_ij) then update the values for qij in the Library
                with the newer one
            }
            else
              Add q_ij to the Library
        }
    }
    Return Library;
}
```

Figure 16: Pseudo code for the algorithm to create common query library.

Conclusions and Future Work

Employing a cognitive user model for IR is a critical and challenging problem to help an IR system to retrieve more relevant documents more quickly. This model accounts for a user's interests and preferences in information seeking and that can be used to determine group collective intelligence with regards to group interests and actions to achieve these interests. Even though there are many interesting published research approaches from both the IR and UM communities that try to address this problem, this work stands out because it fills in the gap between two views: system-focus view from IR and the user-focus view from UM. This work also contributes to the computational collective intelligence community by offering a framework to capture an individual's interests, context and preference, which will lead to ways to improve diversity in a collaborative group. In this summary, we recap the key contribution of this work, and then present different ways of using this framework to infer collective intelligence for a group.

The key contribution is the methodology to construct a hybrid user model. Information about a user and information about an IR system are combined in a decision theoretic framework in order to maximize the effectiveness of a user in an information seeking task with regards to their current searching goals. The hybrid user model truly compensates for the lack of information about a user of an IR system by capturing user intent and adapting the IR system to a user. It also compensates for the lack of information about an IR system by using the information about collections to predict the effectiveness of adaptations. This has never been done before in either the IR or UM communities. By doing so, we allow a user to influence, at a deeper level, the IR system rather than just through the query. Above all, we can use the results of the existing research in IR, IF and UM within a decision theoretic framework to predict the effectiveness of the retrieval task.

The solution presented in this chapter is (i) to convert this problem into a multi-attribute decision problem in which a set of attributes is constructed by compiling the set of attributes describing a user's intent and the set of attributes describing an IR system; and, (ii) to take advantage of research results from the IR community on predicting query performance and from the IF community on determining dissemination threshold to determine sub value functions for these chosen attributes.

One important consequence of this methodology is that this framework can be applied to any other user models for IR or different types of target applications in the same logic. We can determine a set of attributes describing a typical user for that target application, and determine a set of attributes describing the target application and the effectiveness function to assess a user's current goal in using the target application. We can convert it to a decision theoretic framework if the domains for these attributes are finite. The elicitation process is domain or application-dependent.

We also contribute to both communities by reusing the collections, metrics from the IR community such as CACM and MEDLINE; and simulating the traditional procedure as well as creating new procedure for evaluating new features offered by this user model. By reusing the metrics, collections, and simulating the traditional procedure, we ensure comparability. By creating a new procedure, we ensure that we can further evaluate the newly added features from this technique.

Even though the framework for the hybrid user model has been created and substantively implemented and tested, additional work still needs to be done towards the goal of fully integrating both the user and system sides. In the current evaluation framework for the hybrid user model, we cannot evaluate the sub value function for threshold because it only takes the first 15 documents for relevancy assessment and the metric is average precision at three point fixed recalls. For evaluating the sub value function over threshold, a usability evaluation is needed so that the interactions between a user and the IR system are recorded. This can be done implicitly by using additional equipment, recording and processing a user's mouse clicks and gazes. Lastly, an experiment to study the behavior of our hybrid model and how it relates to the certain characteristics of a document collection is planned.

Lastly, this user modeling approach can help address the black-box problem and non-deterministic behavior problem encountered by the CCI community as pointed out in (Szuba, 2001). Additionally, it provides the necessary information to determine and evaluate collective knowledge and collective strategy based on individual knowledge. The specific contributions of this modeling approach to the CCI community are threefold. First, we can use the information captured for each user’s interests, context and actions to create an effective collaborative group in which people share the same common goal but may take different courses of actions and perspectives to solve a problem. We have shown how a common knowledge base can be created from a set of a user’s context network. We can apply a similar principle for creating a common knowledge base of actions. From the processes of creating these knowledge bases, we can identify the users with similar and different interests and actions. Such information helps the process of forming a collaborative group. Second, we can also reason about a group’s interests and context from a given set of user models. Finally, having information about each individual user helps to create testbeds to evaluate and answer the question of whether the collective intelligence is actually helping people do a more effective and efficient job.

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.

Baeza-Yates, R. & Ribiero-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley.

Balabanovic, M. (1998). Exploring versus Exploiting when Learning User Models for Text Recommendation. *User Modeling and User-Adapted Interaction*, 8(1-2), 71–102.

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

Barla, M., & Bieliková, M. (2009). On Deriving Tagsonomies: Keyword Relations coming from the Crowd. In LNAI 5796, N.T. Nguyen, R. Kowalczyk, and S.M. Chen (Eds.), *Proceedings of International Conference on Computational Collective Intelligence, ICCCI 2009,* (pp. 309–320), Wroclaw, Poland.

Belkin. N. J. (1993). Interaction with text: Information retrieval as information seeking behavior. *Information retrieval*. 10. von der Modelierung zur Anwerdung. Konstanz, Germany: Universitaetsverlag, (pp. 55–66).

Belkin, N. J., Oddy, R. N., & Brooks, H. M. (1982). Ask for information retrieval: part I: background and theory. *Journal of Documentation,* 38(2), 61–71.

Belkin, N. J. & Windel, G. (1984). Using monstrat for the analysis of information interaction. In *IRFIS 5, Fifth International Research Forum in Information Science,* (pp. 359–382). Heidelberg.

Billsus, D. & Pazzani M. J. (2000). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3), 147–180.

Borlund, P. (2003). The Concept of Relevance in Information Retrieval. *Journal of the American Society for Information Science and Technology,* 54(10), 913–925.

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

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), 3-10.

Brown, S. M. (1998). *Decision Theoretic Approach for Interface Agent Development.* PhD thesis, Air Force Institute of Technology.

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

Carmagnola, F., Vernero, F., & Grillo, P. (2009). SoNARS: A Social Networks-Based Algorithm for Social Recommender Systems. In *Proceedings of the User Modeling, Adaptation, and Personalization conference,* (pp. 223-234), Trento, Italy.

Cecchini, R. L., Lorenzetti, C. M., Maguitman, A. G., & Brignole, N. B. (2008). Using genetic algorithms to evolve a population of topical queries. *Information Processing and Management,* 44(6), 1863-1878.

Chen, S. Y., Magoulas, G. D., & Dimakopoulos, D. (2005). A flexible interface design for Web directories to accommodate different cognitive styles. *Journal of the American Society for Information Science and Technology,* 56(1), 70–83.

Cooper, W. & Maron, M. E. (1978). Foundations of Probabilistic and Utility Theoretic Indexing. *Journal of the Association for Computing Machinery,* 25(1), 67-80.

de Campos, M. L., Fernández-Luna, M. J., Huete, F. J., & Rueda-Morales, A. M. (2009).

Managing uncertainty in group recommending processes. *User Modeling and User-Adapted Interaction*. 19(3), 207-242.
Drucker, H., Shahrary, B., & Gibbon, C. (2002). Support vector machines: relevance feedback and information retrieval. *Information Processing and Management,* 38(3), 305–323.
Ducheneaut, N., Partidge, K., Huang, Q., Price, B., Roberts, M., Chi, E. H., Belotti, V. & Begole, B. (2009). Collaborative Filtering Is Not Enough? Experiments with a Mixed-Model Recommender for Leisure Activities. In *Proceeding of the seventeenth International conference, User Modeling, Adaptation, and Personalization,* (pp. 295–306), Trento, Italy.
Duong H. T., Uddin, M. N., Lim D., & Jo, G. (2009). A Collaborative Ontology-Based User Profiles System. *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*. 540-552.
Efthimis, E. N. (1996). Query Expansion. In *Williams, M., Eds. 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.
Gaudioso, E., Soller, A., & Vassileva, J. (2006). Preface to the special issue on user modeling to support groups, communities and collaboration. *User Modeling and User-Adapted Interaction.* 16 (3), 171-174.
He, B., & Ounis, I. (2004). Inferring query performance using pre-retrieval predictors'. In *Information Systems, Special Issue for the String Processing and Information Retrieval: 11th International Conference,* (pp. 43–54).
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.
Jensen, F. V. (1996). *An Introduction to Bayesian Networks.* University College London Press, London.
Keeney, L. R. & Raiffa, H. (1976). *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons.
Kim, J. (2009). Describing and Predicting Information-Seeking Behavior on the Web. *Journal of the American Society for Information Science and Technology,* 60(4), 679–693.
Kumaran, G., & Allan, J. (2008). Adapting information retrieval systems to user queries. *Information Processing and Management*, 44(6), 1838–1862
Lau, T. & Horvitz, E. (1999). Patterns of Search: Analyzing and Modeling Web Query Refinement. In *Proceedings of the Seventh International Conference on User Modeling,* (pp. 119-128), Banff, Canada.
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.
Logan, B., Reece, S., & Sparck, J.(1994). Modeling information retrieval agents with belief revision. In *Proceedings of the Seventeenth Annual ACM/SIGIR Conference on Research and Development in Information Retrieval*, (pp. 91–100).
Lopér-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.
Lynch, C. (1992). The next generation of public access information retrieval systems for research libraries: lessons from ten years of the MELVYL system. *Information Technology and Libraries.* 11(4), 405–415.
Masthoff, J.. (2004). Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction*. 14(1), 37-85.
McDoland, D. W., & Ackerman, M. S. (2000). Expertise Recommender: A plexible recommendation system and architecture. In *Proceedings of the CSCW' 00*. (pp. 231-240). Philadelphia, PA.
Michard, M. (1982). Graphical presentation of Boolean expressions in a database query language: design notes and an ergonomic evaluation. *Behaviour and Information Technology*, 1(3), 279–288
Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM,* 38 (11), 39-41.
Nguyen, H. (2005). *Capturing User Intent for Information Retrieval.* Ph.D Dissertation, University of Connecticut.
Nguyen, H., & Haddawy, P. (1999). The Decision-theoretic Interactive Video Advisor. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence- UAI 99,* (pp. 494-501), Stockholm, Sweden
Nguyen, H.., Santos Jr. E., Schuet, A., & Smith, N. (2006). Hybrid User Model for Information Retrieval. *Technical Report of Modeling Others from Observations workshop at Twenty-First National Conference on Artificial Intelligence (AAAI-06) conference,* (pp. 61–68), Boston, MA.
Nguyen, H., Santos, E. J., 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. 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.
Nguyen, N. T. (2008). Keynote Speech: Computational Collective Intelligence and Knowledge Inconsistency in Multi-agent Environments. *Intelligent Agents and Multi-Agent Systems*. (pp. 2-3).
Nguyen, N. T., Katarzyniak, R., Adam, J., (Eds.). (2009). *New Challenges in Computational Collective Intelligence.* Series: Studies in Computational Intelligence, Vol. 244.
Page, S., E. (2007). *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*. Princeton University Press.
Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.
Pogacnik, M., Tasic, J., Meza, M., & Kosir, A. (2005). Personal Content Recommender Based on a Hierarchical User Model for the Selection of TV Programmes. *User Modeling and User-Adapted Interaction,* 15(5), 425-457.
Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science*, 3, 329-354.
Rich, E. (1983). Users are individuals: individualizing user models. *International Journal of Man-Machine Studies,* 18, 199-214.
Rochio, J. J. (1971). Relevance feedback in information retrieval. *The Smart 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. & Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2), 95-145.
Ruthven, I., Lalmas, M., & van Rijsbergen, K. (2003). Incorporating user search behavior into relevance feedback. *Journal of the American Society for Information Science and Technology,* 54(6), 529–549.
Salton, G. & C. Buckley. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4), 288–297.
Santos, E. J., & Nguyen, H. (2009). Modeling Users for Adaptive Information Retrieval by Capturing User Intent. In *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling*. (Edited by Max Chevalier, Christine Julien, Chantal Soulé) IGI Global, 88-118.
Santos, E. J., Nguyen, H., & Brown, S. M. (2001). Kavanah: An active user interface for Information Retrieval Application. In *Proceedings of 2nd Asia-Pacific Conference on Intelligent Agent Technology,* (pp. 412–423), Japan.
Santos, E. J., Nguyen, H., Zhao, Q., & Pukinskis, E. (2003a). Empirical Evaluation of Adaptive User Modeling in a Medical Information Retrieval Application. In *Proceedings of the ninth User Modeling Conference*, (pp. 292–296), Johnstown, PA.
Santos, E. J., Nguyen, H., Zhao, Q., & Wang, H. (2003b). User Modeling for Intent Prediction in Information Analysis. In *Proceedings of the 47th Annual Meeting for the Human Factors and Ergonomics Society (HFES-03),* (pp. 1034–1038), Denver, CO.
Santos, E. J., Zhao, Q., Nguyen, H., & Wang, H. (2005). Impacts of User Modeling on Personalization of Information Retrieval: An evaluation with human intelligence analysts. In *Technical report of Workshop on Evaluation of Adaptive Systems at UM 2005,* (pp. 27-36), Edinburgh, UK.
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 sixth International Conference in User Modeling - UM 97*, (pp. 43–54).
Schmitt, C., Dengler, D., & Bauer, M. (2003). Multivariate preference models and decision making with the MAUT Machine. In *Proceedings of the ninth International Conference in User Modeling – UM03. P. Brusilovsky, A. Corbett, and F. de Rosis (Eds.). Springer.* (pp. 297-302).
Sleator, D. D. & Temperley, D. (1993). Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, (pp. 277–292).
Spink, A. & Cole, C. (2005). *New Directions in Cognitive Information Retrieval*. The Information Retrieval Series. Springer.
Spink, A. and Losee, R. M. (1996). Feedback in information retrieval. In Williams, M., Eds., *Annual Review of Information Science and Technology*, 31, 33–78.
Spink, A., Greisdorf, H., & Bateman, J. (1998). From highly relevant to not relevant: Examining different regions of relevance. *Information Processing and Management.* 34(5). 599–621.
Szuba M. T. (2001). Computational Collective Intelligence. John Wiley & Sons, Inc.
Upton, K., Kay, J. (2009). Narcissus: Group and Individual Models to Support Small Group

Work. In *Proceedings of the User Modeling, Adaptation, and Personalization conference,* (pp. 54-65), Trento, Italy.

Vickery, A. & Brooks, H. (1987). Plexus: The expert system for referral. *Information Processing and Management*, 23(2), 99–117.

Voorhees, M. E. (2008). On test collections for adaptive information retrieval. *Information Processing and Management*, 44(6), 1879-1885.

Xie, Y. & Raghavan, V. V. (2007). Language-Modeling Kernel Based Approach for Information Retrieval. *Journal of the American Society for Information Science and Technology*, 58(14), 2353-2365.

Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV Program Recommendation for Multiple Viewers Based on user Profile Merging. *User Modeling and User-Adapted Interaction,* 16(1), 63-82.

Zanker, M., & Jessenitschnig, M. (2009). Case-studies on exploiting explicit customer requirements in recommender systems. *User Modeling and User-Adapted Interaction,* 19(1-2), 133-166.

Zhang, Y. (2008). Complex adaptive filtering user profile using graphical models. *Information Processing and Management,* 44(6), 1886–1900.

Zhao, Q., Santos, E.,J., Nguyen, H., & Mohammed, M. (2009). What makes a good summary? *Computational Methods for Counterterrorism.* Edited by Shlomo Argamon, Newton Howard. Spinger, 33-50.