

# Evaluation of Effects on Retrieval Performance for an Adaptive User Model

Hien Nguyen, Eugene Santos Jr., Qunhua Zhao, and Chester Lee

Computer Science and Engineering Department  
University of Connecticut  
191 Auditorium Road, U-155, Storrs, CT 06269-3155  
{hien,eugene,qzhao,clee}@cse.uconn.edu

**Abstract.** One of the challenging problems for evaluating the effectiveness of a user model with regards to retrieval performance is the absence of an evaluation method that offers the ability to compare with other existing approaches while assessing the new features offered by a user model. In this paper, we report our method of using collections, procedures and metrics from the information retrieval community to evaluate a cognitive user model which captures user intent to improve retrieval performance and adapts to a user's interests, preferences and context. Specifically, by starting with an empty user model for each query, we simulate the process of assessing the short-term effects of relevance feedback techniques in traditional information retrieval. By using a seed user model learned from relevance feedback, we assess both short and long-term effects on the entire search session. In this paper, we show how we can compare user modeling approaches by using the above method against a classic information retrieval approach, the Ide dec-hi, using CACM and Medline collections. This evaluation also helps analyze and address the strengths and weaknesses of our model and develops appropriate solutions.

## 1 Introduction

One of the challenging problems with evaluating an adaptive user model for information retrieval (IR) is the absence of an evaluation method that offers the ability to compare with other existing approaches in IR community while accessing the new, special features brought by the model. In the user modeling (UM) community, many researchers have explored the use of user models for improving retrieval performance [1, 2] and have evaluated the effectiveness of their user models on retrieval performance by using their own collections, tasks and procedures. Therefore, it is very difficult to compare them against different techniques, especially against the techniques used in the IR community. On the other hand, standard metrics, collections and procedures have been established and used in the IR community for decades to evaluate different retrieval techniques, especially the techniques that use relevance feedback (RF) and query expansion (QE) to improve retrieval performance [13]. However, the user model

created by using IR techniques such as RF and QE is short-lived. The model only affects a single query instead of the entire search session. In addition, these techniques assessed the retrieval performance in isolation. Therefore, using these procedures alone may not fully evaluate the special features that are created by long-lived user models.

In this paper, we report our evaluation with regards to retrieval performance for a cognitive user model which captures user intent for IR [14–16]. This is one important phase of an ongoing three-phase evaluation proposed in [10] in which we evaluate the correctness of the process of capturing user intent, the effectiveness of the user model on retrieval performance and user performance. The power of our method lies with its objectivity, inexpensiveness and comparability. The objectivity is reflected in using the collections and metrics, which do not depend on a particular set of users nor a set of parameters used in the adaptive system being tested. The procedures are lightweight and can be used for any other adaptive system in information retrieval. The comparability is achieved by using a standard procedure as a part of our evaluation, in which we simulate the traditional procedures used in the IR community. We seek to address two important questions: (1) How can we use collections, metrics and procedures from the IR community to evaluate our user model, especially its short-term and long-term effects? and (2) How does this evaluation help us analyze the overall effectiveness of our user model on user and system performance? Our user model captures user intent dynamically by analyzing information in retrieved relevant documents. Therefore, we compare our approach with the best traditional approach for RF, the *Ide dec-hi* approach using term frequency inverted document frequency weighting scheme (TFIDF) [13] on CACM and Medline collections.

This paper is organized as follows: We begin with a review of some important related work in IR and UM communities regarding the evaluation of a user model for IR. Next, we briefly present our approach. Then our evaluation method is presented, followed by the analysis of the results and our discussion. Finally, we present our conclusions and future work.

## 2 Related work

The main problems for evaluating the effectiveness of a user model for IR in terms of retrieval performance lie with the difficulty in comparing different approaches and the limitation of using this result to improve user performance. These problems are the results of little overlap between IR and UM communities when building user models for IR, as identified in [17].

In the IR community, user models have been created by using IR techniques such as RF and QE [18]. IR researchers have developed metrics, procedures and collections to assess the effectiveness of these two approaches for decades. Specifically, in the study done by Salton and Buckley [13], a common evaluation framework has been laid out to evaluate twelve different RF techniques, including *Ide dec-hi*. This framework offers the ability to compare different techniques with each other by using average precision at three specific recall points (0.25, 0.5 and

0.75) (we call this *three point fixed recall*). It also ensures that we assess a RF technique based on *new* information retrieved by using residual collections. A residual collection is created by removing all documents previously seen by a user from the original collection regardless of whether they are relevant or not; then the evaluation process is done using the reduced collection only. Some other techniques, such as freezing and test/control groups, are used to evaluate RF and QE techniques [20].

In the UM community, the common practice is to perform experiments with a particular set of users with and without the presence of a user model [5]. While this process is definitely needed to evaluate any adaptive system, it is expensive and the result depends on the particular group of users who participated in the experiments. Therefore, in order to better prepare for the experiments with real users, it is a good idea to first test a system within a simulated environment. By combining the results in the simulated and real environments, we can further analyze the outcomes from different perspectives. So far, most of the studies [1–3, 9] use two common metrics in IR: precision and recall [12]. These studies, unfortunately, use their own test collections and tasks; thus making any comparison difficult for current and future approaches.

### 3 Our user modeling approach

In our model, we capture, represent and use the information on *what* a user is currently interested in (the Interests); *how* a query needs to be constructed (the Preferences) and *why* the user dwells on a search topic (the Context) to modify a user’s queries pro-actively.

#### *Our user model architecture*

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* ( $P$ ). While previous efforts have focused on capturing just a single aspect, none of them have combined these three aspects in capturing user intent. A context network ( $C$ ) is a directed acyclic graph (DAG) that contains *concept nodes* and *relation nodes*. Concept nodes are noun phrases representing the concepts found in retrieved relevant documents (e.g. “*computer science*”). Relation nodes represent the relations among these concepts. There are two relations captured: set-subset (“*isa*”) and relate-to relations (“*related to*”). We construct  $C$  dynamically by finding a set of subgraphs in the intersection of all retrieved relevant documents. Each document is represented as a *document graph* (DG), which is also a DAG. We developed a program to automatically extract DG from text. We extracted noun phrases (NPs) from text using Link Parser [19]; these NPs will become concept nodes in a DG. The relations nodes are created by using three heuristic rules: *noun phrase heuristic*, *noun phrase-preposition phrase heuristic*, and *sentence heuristic*.

Each node in  $C$  is associated with its *weight*, *value* and *bias*, which are used by a spreading activation algorithm to reason about the new set  $I$ . In this algorithm, a concept that is located far from an observed interest concept will be of less

interest to the user. After we find the set of common subgraphs, we check to see if a subgraph is not currently in  $C$ , and add it accordingly. We ensure that the update will not result in a loop in  $C$ . As we can see from the representation of  $C$ , which contains the relations between concept nodes which represent potential goals that a user wants to explore. Therefore, it can be used to explain why a user is particularly interested in this concept based on its relations with more general/more specific/or related concepts.

Each element of  $I$  consists of an *interest concept* ( $a$ ) and an *interest level* ( $L(a)$ ). An interest concept refers to a concept a user is focusing on, and an interest level is any real value from 0 to 1 representing how much emphasis the user places on a concept. Based on the values of each interest concept found in  $C$ , we produce a rank ordering of the concepts to build  $I$ . Since a user's interests change over time, we incorporate a fading function to make the likely irrelevant interests fade away. We compute  $L(a)$  after every query by:  $L(a) = 0.5 * (L(a) + \frac{n}{m})$  with  $n$  as the number of retrieved relevant documents and  $m$  as the number of retrieved documents containing this concept  $a$ . If  $L(a)$  falls below a threshold,  $a$  is removed from  $I$ .

We use a Bayesian network [7] to represent  $P$  because of its expressiveness, and ability to modeling uncertainty. There are 3 kinds of nodes in  $P$ : pre-condition ( $Pc$ ), goal ( $G$ ), and action ( $A$ ) nodes. A user's query and the concepts contained in  $I$  are examples of  $Pc$ . An example of  $G$  is a tool called a filter that narrows down the search topics semantically or an expander that expands the search topics semantically. An example of  $A$  is a modified query. For each pre-condition node representing a user's current interest, its prior probability will be set as the interest level of the corresponding interest concept. The conditional probability table of each goal node is similar to the truth table of logical AND. Each  $G$  is associated with only one  $A$ . The probability of  $A$  is set to 1 if the tool is chosen and to 0, otherwise.

$P$  is updated when a user gives feedback. The preference network adapts based on the observation of interactions between a user and our system. Two new preference networks are created; one of them contains a new tool labelled *filter*, and another contains a new tool labelled *expander*. The correction function calculates the probability of a new network that improves the user's effectiveness for both of the two new preference networks. The preference network is updated according to the one with higher probability. The calculation is used to determine the frequency that a tool helps in the previous retrieval processes. If the total number of retrieved relevant documents exceeds a threshold, the tool is considered helpful.

#### *Integrating our user model into an IR system*

- Given a user model  $M=\{I, P, C\}$  and a query graph (QG)  $q$ . A QG is similar to a DG but is built from a user's query.
- Re-compute the values of interest concepts found in  $C$  by a spreading activation algorithm on  $C$  to construct  $I'$ .
- We set as evidence all interest concepts of  $I'$  found in  $P$ . Find a pre-condition node  $Pc$  representing a query in  $P$  which has associated query graph(QG)

that completely or partially matches against the given  $q$ . If such a node  $Pc$  is found, set it as evidence.

- Perform belief updating on  $P$ . Choose top  $n$  goal nodes from  $P$  with highest probability values. We call this set of goals as  $SG$ .
- For every goal node  $g$  in  $SG$ : If the query has been previously submitted and the user has used  $g$ , replace the original query subgraph with the graph associated with the action node of this goal. If the query has not been submitted before and  $g$  represents a filter: For every concept node  $q_i$  in the user’s query graph  $q$ , we search for its corresponding node  $cq_i$  in  $C$ . For every concept  $a_i$  in  $I$ , we search for its corresponding node  $ca_i$  in  $C$  such that  $ca_i$  is an ancestor of  $cq_i$ . If such  $ca_i$  and  $cq_i$  are found, we add the paths from  $C$  between these two nodes to the modified query graph. It works similarly with an expander except that  $ca_i$  should be a progeny of  $cq_i$ .

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

## 4 Evaluation method

### 4.1 Overview

The goal of our evaluation method is two-fold. First, it offers the ability to compare with the existing approaches by using standard collections, metrics and procedures from the IR community. We compare our approach against the Ide dec-hi with TFIDF, therefore, the procedures used for evaluating these techniques must be adhered. Second, our evaluation method assesses the special feature of our user model, which is the use of knowledge learned over time to modify queries. The procedure, therefore, needs to assess the effects of the users’ prior knowledge and combination between the users’ prior knowledge and knowledge learned from a query or a set of queries.

### 4.2 Testbeds

We chose CACM and Medline as our testbed collections because they have been widely used for evaluating the effectiveness of some important RF and QE techniques [13, 4, 8]. CACM contains 3204 documents and 64 queries in computer science and engineering (CSE) domain while Medline contains 1033 documents and 30 queries in the medical domain. The characteristics of the CACM and Medline documents used in our evaluation are shown in Tables 1. We evaluated our user model and TFIDF with Ide-dec hi approach over the entire set

of questions from these two collections because we wanted to obtain a baseline performance for our approach on these two collections.

<i>Attributes</i>	CACM	Medline
Total vectors	3204	1033
Mean length of vector	19.57	53.36
Standard deviation of length of vector	21.91	24.83
Mean frequency of term in a vector	1.61	1.46
Percentage of term with frequency 1	89%	74.78%

**Table 1.** Characteristics of CACM and MEDLINE documents

### 4.3 Procedures

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

We follow the procedure laid out by Salton and Buckley [13]. For the Ide dec-hi/TFIDF, each query in the testbeds is converted to a query vector. The query vector is compared against each document vector in the collection. For our approach, we construct a QG for each query in the testbeds in the same way that we construct DGs, in which Link Parser [19] is used. Link Parser sometimes produces incorrect parse trees for the sentences with words which are not found in its dictionary. Therefore, we manually created 27 QGs out of 30 queries for Medline and 21 QGs out of 64 queries for CACM. Medline contains many specialized terms used in the medical domain and CACM contains many specialized terms used in the CSE domain which are not found in Link Parser’s dictionary. We would like to make sure that we have correct QGs to work with. There is no intervention made during the construction of DGs. The main difference between vector representation of TFIDF and our graph representation described above is that the former focuses on frequency of an individual word while ours focuses on the relationship among terms. After we issue each query, the relevant documents found in the first 15 returned documents are used to modify the original query. For the Ide dec-hi/TFIDF, the weight of each word in the original query is modified from its weights in relevant documents and the first non-relevant document. The words with the highest weights from relevant documents are also added to the original query. For our user modeling approach, we start with an empty user model and add the concept and relation nodes to the original QG based on the procedure described in Section 3. We then run each system again with the modified query. We refer to the first run as *initial run* and the second run as *feedback run*. For each query, we compute average precision at three point fixed recall (0.25, 0.5 and 0.75).

#### *Special procedure for user modeling approach*

The special procedure assesses the effects of prior knowledge and the combination of prior knowledge with knowledge learned from a query or a group of queries. These requirements lead to our decision to perform 4 experiments:

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

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

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

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

In this procedure, we use the prior knowledge, which is dynamically constructed after Experiment 1 as opposed to using no prior knowledge as in the standard procedure above.

## 5 Results and Discussions

### 5.1 Results for standard procedure

The average precision at three point fixed recall of the initial run and feedback run using residual collection of the experiments in standard procedure for CACM and Medline is reported in Table 2. Those in previous publications achieved a slightly better results compared to ours because (i) we used the entire set of queries, while others, for example [4] used a subset of queries; and (ii) we treat the terms from title, author and content equally. Table 2 shows that we achieved competitive performance in both runs for residual and original collections.

The results of our special procedure on user modeling approach are shown in Table 3. Experiment 2 shows that by using the seed user model as prior knowledge for a user, the precision has been increased for the initial runs. Experiments 1, 3 and 4 show that by using our user model, the precision of the feedback runs is always higher using residual and original collections than those of the initial runs. For both collections, we can see that among the four experiments, Experiment 4 performs competitively compared to Ide dec-hi in the feedback run while it offers the advantages of having higher precision in the initial run compared to TFIDE. This shows that we have already retrieved quality documents earlier in the retrieval process than the other approach, leaving less relevant documents

	TFIDF/Ide dec-hi		User modeling	
	Residual	Original	Reisidual	Original
<b>CACM</b>				
Initial run	0.065	0.091	0.067	0.095
Feedback run	0.12	0.2	0.090	0.223
<b>MEDLINE</b>				
Initial run	0.19	0.39	0.212	0.4
Feedback run	0.32	0.54	0.328	0.583

**Table 2.** Average precision at three point fixed recall for standard procedure

Experiments	CACM		Medline	
	Residual	Original	Residual	Original
Exp 1.				
Initial run	0.073	0.095	0.212	0.446
Feedback run	0.091	0.223	0.344	0.614
Exp 2.				
Initial run	0.075	0.095	0.249	0.512
Exp 3.				
Initial run	0.075	0.095	0.249	0.512
Feedback run	0.11	0.21	0.343	0.609
Exp 4.				
Initial run	0.082	0.095	0.258	0.525
Feedback run	0.11	0.23	0.360	0.625

**Table 3.** Average precision at three point fixed recall for special procedure

for us to retrieve in the feedback run. The average precisions of experiments 3 and 4 (in which seed user models are used) are higher than those of experiments 1 and experiments in standard procedure for both collections most of the time.

## 5.2 Discussion

The standard procedure offers us a chance to compare with the TFIDF and Ide dec-hi approaches using their evaluation procedures on the same collections. These queries are as complicated as the ones asked by any real user. However, the evaluation procedures is lightweight and they can be easily used to evaluate adaptive systems before hiring the real subjects. This maintains objectivity and serves as a baseline comparison for future extensions. The special procedure evaluates the long-term effects of knowledge learned in three ways: (i) using the seed user model as prior knowledge, (ii) using the seed user model and updating it with knowledge learned from a query only, and (iii) using the seed user model and updating it with knowledge learned again from a set of queries.

The results show that the retrieval performance increased with all three of these methods. This methodology shows the best performance using a combination of prior knowledge and knowledge learned from a group of queries. For example, in Experiment 4 of the special procedure on Medline, question 7 in the initial run has an added relation “*radioisotop scan - isa - scan*” by the user model and thus has retrieved two more relevant documents in the top 15 than it did in Experiments 1, 2 and 3 (6 relevant documents in the top 15 in Experiment 4 vs 4 relevant documents in top 15 in Experiments 1,2, and 3). We have also applied this method to another collection CRANFIELD [11], and show that our user modeling approach has the potential to improve efficiency, learnability, and interactivity between a user and an IR system by retrieving more highly relevant documents, quickly. Our work here demonstrates this evaluation methodology can be used to assess the impact of knowledge captured by our user model over time to IR process.

## 6 Conclusion

In this paper, we have reported our evaluation method to assess the effectiveness of our user model with regards to retrieval performance using CACM and Medline collections. The results of this evaluation show how we can compare the user modeling approaches using procedures, collections and metrics of the IR community while still being able to assess special features of the models.

There are issues that we wish to address from this research. Our user modeling approach works best if a user has demonstrated his/her searching styles. So, we will consider re-ordering the queries to effect different search styles (e.g users explore a topic, its subtopics, and then change to a new topic). It will help closely relate the experiment to real life situations while maintaining its objectivity. In this current evaluation, we used the seed user model obtained from Experiment 1. In the future, the seed user model can be created manually (which is likely to achieve even better results) or can be learned from a training query set.

We would like to combine the results of this phase with two other phases [10] to provide a big picture analysis of the overall effectiveness of our user model. This evaluation experiment plays a very important role in the analysis of the overall effectiveness of our user model in terms of improving retrieval and user performance. This data gives us the relevant documents identified by experts who created these collections while the data from our assessments of user performance will give us the relevant documents identified by real users with varying levels of expertise. We will then be able to draw a clear connection between objective and subjective relevancy and how they affect the retrieval performance as well as user performance.

## References

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

2. Billsus, D., Pazzani, M.J.: User modeling for adaptive news access. *Journal of User Modeling and User-Adapted Interaction* **10** (2000) 147–180
3. Bueno, D., David, A.A.: Metiore: A personalized information retrieval system. In: Bauer, M., Vassileva, J. and Gmytrasiewicz, P. (Eds.). *User Modeling: Proceedings of the Eight International Conference, UM2001, Berlin, Springer* (2001) 168–177
4. Loper-Pujalte, C., Guerrero-Bote, V., Moya-Anegon, F.D.: Genetic algorithms in relevance feedback: a second test and new contributions. *Information Processing and Management* **39(5)** (2003) 669–697
5. Chin, D. Evaluating the effectiveness of user models by experiments. Tutorial at User Modeling conference, Johnstown, Pittsburgh. (2003).
6. Frake, W.B., Baeza-Yates, R.: *Information Retrieval: Data Structures and Algorithms*. Prentice Hall PTR, Upper Saddle River, NJ 07458 (1992)
7. Jensen, F.V.: *An Introduction to Bayesian Networks*. Univ. College London Press, London (1996)
8. Drucker, H., Shahraray, B., Gibbon, C.: Support vector machines: relevance feedback and information retrieval. *Information Processing and Management* **38(3)** (2002) 305–323
9. Magnini, B., Strapparava, C.: Improving user modeling with content-based techniques. In: In Bauer, M, Vassileva, J, and Gmytrasiewicz, P. (Eds). *User Modeling: Proceedings of the Eighth International Conference, UM2001, Berlin, Springer* (2001) 74–83
10. Nguyen, H.: Capture user intent for information retrieval. In: *Doctoral Consortium at AAAI 2004*. (2004) To appear.
11. Nguyen, H., Santos E. Jr., Zhao, Q. and Wang, H. Capturing User Intent for Information Retrieval. In: *the Proceedings of the 48th Annual meeting for the Human Factors and Ergonomics Society (HFES-04)*. October 2004, New Orleans. (2004). To appear.
12. Salton, G., McGill, M.: *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company (1983)
13. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* **41(4)** (1990) 288–297
14. Santos, E., Nguyen, H., Brown, S.M.: Kavanah: An active user interface information retrieval application. In: *Proceedings of 2nd Asia-Pacific Conference on Intelligent Agent Technology*. (2001) 412–423
15. Santos, E. Jr., Nguyen, H., Zhao, Q., Hua, W.: User modelling for intent prediction in information analysis. In: *Proceedings of the 47th Annual Meeting for the Human Factors and Ergonomics Society (HFES-03)*. (2003) 1034–1038
16. Santos, E. Jr., Nguyen, H., Zhao, Q., Pukinskis, E.: Empirical evaluation of adaptive user modeling in a medical information retrieval application. In: *Proceedings of the ninth User Modeling Conference*. (2003) 292–296 Johnstown. Pennsylvania.
17. Saracevic T., Spink A., Wu W. Users and Intermediaries in Information Retrieval: What Are They Talking About? *Proceedings of the 6th International Conference in User Modeling* (1997) 43–54 Springer-Verlag Inc.
18. Spink A., and Losee R. M. Feedback in information retrieval. Williams, M., ed., *Annual Review of Information Science and Technology* **31** (1996) 33–78
19. Sleator, D.D., Temperley, D.: Parsing english with a link grammar. In: *Proceedings of the Third International Workshop on Parsing Technologies*. (1993) 277–292
20. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review* **18** (2003)