Formal Methods for Evaluating Information Retrieval in Hypertext Systems

Yuri Quintana, Mohamed Kamel, Rob McGeachy
Pattern Analysis and Machine Intelligence Group
Department of Systems Design Engineering
University of Waterloo
Waterloo, Ontario, Canada
< yquint, mkamel > @watnow.UWaterloo,ca

ABSTRACT

One common method for organizing and retrieving information is hypertext. Hypertext users can search for information by selecting association links from one item to another, following an association trail towards the information they seek. This paper examines the nature of association searches in hypertext systems and proposes a formal model that can be used to evaluate information retrieval from hypertext documents. One of the main motivations for the development of a hypertext search model is that it can be used to influence the early design of a system or document. For example, each time a new design of a hypertext system or document is created, the model could be used to estimate the information retrieval time of users. The model we present can be used to determine the time needed to find specific information in hypertext documents, based on the structure of the document, the experience level of the user, and the design of the hypertext system's user interface. A preliminary set of empirical observations is described that reveals the strengths and limitations of our model.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1993 ACM 0-89791-630-1/93/0010/0259 $1.50

1 INTRODUCTION

One common method for organizing and retrieving information is hypertext [1, 7, 21, 25]. The main idea of hypertext is the storage of information as objects or nodes that are interconnected with association links. Hypertext users can enhance their perception and understanding of vast amounts of information by examining the association links between information items. Hypertext has been shown to be a useful tool for browsing [6, 17] and learning [9]. Hypertext systems can also be used to retrieve specific information [12, 18] since users can follow an association trail from one item to another until the desired information is found. Studies [11, 26] have shown that users can become disoriented and lost while navigating through large, highly interconnected hypertext documents [7]. This paper examines the nature of association searches in hypertext systems and proposes a formal model that can be used to evaluate information retrieval from hypertext documents.

1.1 Motivation for A Formal Predictive Model

One of the main motivations for the development of a hypertext search model is that is can be used to influence the early design of a system. For example, each time a new design of a hypertext system or document is created, a predictive model could be used to estimate the information retrieval performance of users. Initial design flaws and inefficiencies can be detected and corrected without the need of a full usability experiment involving recorded observation of users, which can be a very time consuming and expensive task. A full usability experiment will still be needed later in the design process to reveal other problems that users may encounter which are not considered by the model. However, the use of predictive models can be a valuable and cost effective way to determine if design changes will actually improve retrieval time for users.

1.2 Organization of the Paper

In this paper we focus on hypertext searching in *closed tasks* [19], where the user's goal is to find the answer to a specific question. We propose a formal predictive model to determine the time needed to find specific information in hypertext documents, based on the structure of the document, the experience level of the user, and the design of the hypertext system's user interface. A preliminary set of empirical observations are described that reveal the strength and limitations of our model.

This paper is organized as follows. Section 2.0 reviews some previous studies on hypertext searching. Section 3.0 describes a framework for modelling hypertext search behaviour and discusses how some perceptual models are related to each other. Section 4.0 describes our formal predictive model for hypertext searching. Section 5.0 discusses some empirical observations and discusses the strengths and limitations of our model. Section 6.0 summarizes the contributions of our work to the field of online documentation and human-computer interaction and discusses some possible extensions to our work.

2 LITERATURE REVIEW

Meta-analysis of empirical hypertext search studies is difficult since previous studies have been carried out in varying conditions and have been reported with varying levels of detail in the literature. Below we describe some behavioural characteristics that have been observed in these studies.

Nielsen [24] surveyed 30 scientific studies on hypertext and identified 92 different aspects of hypertext design that have been evaluated and compared. Nielsen's objective was to normalize the results and identify the most important design issues. Nielsen concluded that the following factors have the most influence on hypertext search behaviour: 1) age, 2) motivation, 3) user preconceptions, 4) level of expertise, and 5) navigation mechanism. Nielsen also concluded that few studies have been large or detailed enough to arrive at statistically significant results. However, it appears that the user's expertise and hypertext system's navigation mechanisms are prominent factors in hypertext search behaviour.

Nielsen [22, 23] has also investigated disorientation in hypertext searching. A study on the usability of videotex systems [22] showed that the backtracking facility was one of the most frequently used navigation facilities, especially for novice users. This observation is similar to the evaluation of the Glasgow Online hypertext system [14] where half the users tended to return to the title page and retrace long routes in cases where backtracking was either not available or difficult to use. Nielsen and Lyngbaek [23] reported on a study where 25 people were asked to read a particular hypertext document for one hour. It was observed that 57% percent of the users indicated they felt they were confused about where they were, and only 22% indicated that they knew how to get back. Also, only 35% indicated that they felt confident that they had found all the information for which they were searching.

Recently, Rouet [28] has analysed some empirical studies of hypertext and analysed the effectiveness of hypertext for text comprehension, learning, and information retrieval. He concludes that there is no consistent evidence for the advantage of hypertext over linear presentation formats. He also concludes that performance on hypertext documents varies according to: 1) subject's expertise, 2) interface features, 3) previous training on the interface, and 4) task requirements.

Below are some other findings of the empirical studies on hypertext searching. Factors 1,2,3 relate to the characteristics of the user, factors 4 and 5 relate to the task definition, and factors 6, 7, and 8 relate to the user interface:

1. The user's motivation is a major factor in search completion times [24].
2. The user's knowledge of and training on the hypertext interface improves searching [19, 28].
3. Users with better spatial visualization ability are less likely to become lost [2].
4. Searching is faster when queries are well defined [19].

5. Searching is faster when the words in the question also appear in the titles and the body of hypertext nodes [9, 11].
6. Navigational interface features [26], particularly overview maps [20], reduce conitive disorientation and improve search times.
7. Many users begin using search methods that are familiar to them from the paper domain and move to new methods as they learn and find them effective [9].
8. Backtracking is one of the most frequently used hypertext interface features [14, 22, 23].

3 A FRAMEWORK FOR MODELLING HYPERTEXT SEARCHING

This section provides a framework for modelling a hypertext user, a hypertext document, a hypertext system, and a set of search tasks. It begins with a description of a real world scenario of hypertext searching, and introduces several models and their interactions. The information that needs to be stored in each model is also defined and related to the cognitive tasks of hypertext searching. This framework provides the basis for our predictive model which is described in the next section.

3.1 Real World Scenario for Hypertext Searching

Searching for information in hypertext involves the interaction of several models of perception from: 1) the author of the hypertext document, 2) the designer of the hypertext application, 3) the evaluator who defines the search tasks, and 4) the searcher under evaluation. Differences in perceptions account for differences in search behaviour. A real world scenario for hypertext searching is shown in Figure 1 illustrating some of the perceptual models and their connections via physical devices.

A user model is the representation of the knowledge that a user has during a problem solving task. The conceptual model is the model of the author of the hypertext document. The author's perception of the domain is reflected in the structure, content, and wording of the hypertext document. This conceptual model is often quite different from the user model of the hypertext searcher. This difference is in part due to the prior knowledge that a user has of the domain, and the way new knowledge is interpreted from the hypertext document and assimilated with previous knowledge.

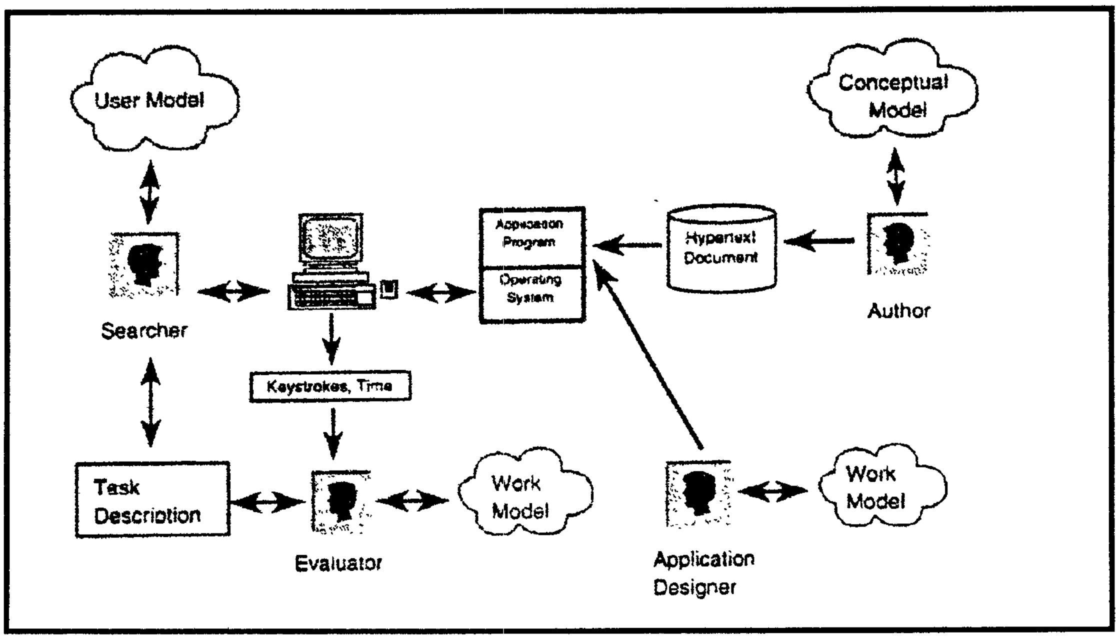

Figure 1: Real World Scenario for Hypertext Searching

The user's interpretation of the document is also affected by the hypertext application and the interaction style or dialog needed to view the hypertext document. The hypertext application designer builds the hypertext system based on a model of how the user will want to work with hypertext documents. However, the work model of the hypertext designer is often different from the way any individual works. This occurs because the designer is building one application for many users, each with his or her own personal preferences and abilities. Thus, the work model of user envisioned by the application designer is often the most common way in which most users are likely to want to interact with the document.

Finally, the evaluator designs a set of tasks that are based on what the evaluator thinks are the most typical tasks of users. This work model can be different from the application designer's work model and different from the way any one individual user is likely to work.

The following section describes some of the basic cognitive tasks that are performed during hypertext searching. These tasks are then related to the information that is stored in the user model, conceptual model, and work models.

3.2 Cognitiive Processes in Hypertext Searching

The most basic cognitive process that a user performs during a search is the selection of a hypertext link. This process is repeated until the desired information is found (also known as a goal). During a search, a user may develop sub-goals and strategies which may be re-formulated periodically.

A user will end the search when: 1) the information is found, 2) the user is taking too long and abandons the search in frustration, or 3) the user abandons the search on the belief that the information is not in the document. This search process can be represented using a set of cognitive tasks as shown in Figure 2.

```
1. Read and interpret the search
   task as described by the evalua-
   tor.

2. Formulate a Search Goal, Sub-
   goals, and a Search Strategy.

3. Read and interpret the text in
   the first node.

4. While ( Goal Not Found )

   4.1 If frustration is greater
   than frustration threshold

        Then end search

   4.2 If user is convinced that
   information is not in document

        Then end search

   4.3 If current path is perceived
   as not closing in on search goal

        Then Reformulate search
   sub-goals
         and search strategy

   4.4 Decide on a Link.

   4.5 Physically Select a Link.

   4.6 Read and interpret the text
   in the new node.

5. End While

6. End Search
```

Figure 2: Cognitive Tasks of Hypertext Searching

Figure 3 shows a user model for hypertext searching. Also shown are the work model of the evaluator, the document model of the author and the interface model of the application designer. These models are described below and related to the cognitive search tasks of Figure 2.

During the course of a search, the user maintains three distinct types of knowledge: 1) task knowledge, 2) domain knowledge, and 3) computing knowledge. Task knowledge is knowledge about how to perform a search task. It represents the interpretation of the goal, the decomposition of the

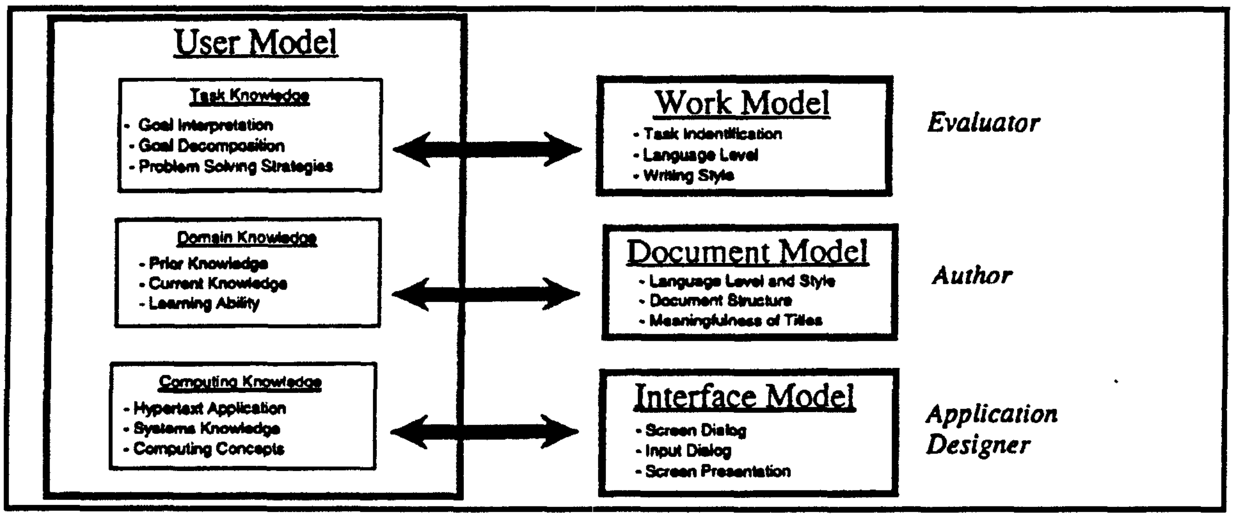

Figure 3: Cognitive Models for Hypertext Searching

goal into sub-goals, and a strategy for achieving these goals. The task knowledge is used in Steps 1, 2, 4.1, and 4.3 shown in Figure 2.

A user's interpretation of the search task may be different from what the evaluator intended. The work model in Figure 3 represents the evaluator's perception of the search tasks. Within this model, task identification refers to the selection of search tasks. Language level and writing style refer to the way in which the search tasks are described. The language level and writing style directly influence the user's interpretation of the search task (Step 1 in Figure 2).

The user model also maintains a representation of the user's interpretation of the domain knowledge. Domain knowledge is the subject or topics described in the hypertext document. The user model shown in Figure 3 represents: 1) the user's knowledge about the domain prior to the viewing the document, 2) the current or recently acquired knowledge, and 3) the learning ability of the user. Current knowledge refers to the knowledge that is learned from viewing the hypertext document. Learning ability is the rate at which new knowledge is extracted from the hypertext document. One aspect not shown is the assimilation of new knowledge with previous knowledge and issues of belief revision. Domain knowledge is used in Steps 3, 4.2, 4.3, 4.4, and 4.6 shown in Figure 2.

The document model shown in Figure 3 represents the author's style in expressing the domain in words. Within this model, language level and style refer to writing technique used in the hypertext document. Document structure refers to the organization of topics in the document. Meaningfulness of titles refers to how well titles reflect the topics in a section. The user's interpretation of new knowledge acquired from a hypertext document can differ from the author's interpretation of that knowledge. The way in which an author expresses knowledge about the domain in words directly influences the acquisition of new knowledge by the user. (Steps 3 and 4.6 in Figure 2).

The user also maintains computing knowledge. Computing knowledge refers to the user's understanding of the hypertext application, the system (hardware and software) on which the application runs, and basic computing concepts (such as the concept of a computer file or directory). This knowledge is used by the user when physically interacting with an interface (Step 4.5 in Figure 2).

The user will perform tasks on the hypertext application based on prior computing knowledge and an interpretation of what is observed in the computer interface. The interface model shown in Figure 3 represents the way the interface appears on the screen and the operations needed to manipulate the document. Screen dialog refers to the basic operating system commands for manipulating the interface such as opening and closing windows, or scrolling within a window. Input dialog refers to the manner in which input commands are performed within the hypertext application for tasks such as selecting links. Screen presentation refers to the size of the screen and the colors used to identify different areas of the screen. The interface can influence both the selection of links (Step 4.5 in Figure 2) and the user's ability to read and comprehend text (Steps 3 and 4.6 in Figure 2).

3.3 Summary of Modelling Hypertext Searching

We have demonstrated how perceptual models from the author of the hypertext document, the designer of the hypertext application, the evaluator, and the searcher interact via physical devices.

The basic cognitive tasks of hypertext searching have been identified and related to these perceptual models. The user's search behaviour has been shown to be influenced by:

1. The user's domain knowledge.
2. The user's knowledge of the hypertext application.
3. The user's knowledge of the computing system.
4. The writing style and organization of the document.
5. The interaction style of the hypertext application.
6. The selection and representation of search tasks.

In the next section we describe a formal method for representing these models for the purpose of predicting information retrieval time from hypertext documents.

4 A FORMAL PREDICTIVE MODEL FOR HYPERTEXT SEARCHING

We first describe objectives of our predictive model and then describe the components of the model. A predictive function is then defined based on these components. We conclude by showing how the predictive model can be used to estimate the cost of a hypertext document in terms of information retrieval time.

4.1 Objectives of the Predictive Model

The objective of the predictive model is to predict the time it takes a user to find a specific item of information in a hypertext document based on the experience level of the user, the structure of the document, and the design of the hypertext system's user interface. The model should be applicable to a range of hypertext systems and documents. Given these objectives, we propose a predictive model that is based on the following components:

- E - A model of the **expertise** of a user.
- U - A model of the **user interface**.
- D - A model of the structure of the hypertext document.
- N_t - The target **node** to be searched for in the hypertext document.
- predict (N_t,D,E,U,Path)- A function that predicts the time it takes a user with expertise E to find node N_t in document D using a hypertext interface U along a given **Path**.

With this model, different hypertext systems can then be compared by varying the model of the user interface U. Different document structures can also be compared by varying the document model D. The retrieval times between varying degrees of user expertise can be compared by varying the model of the user E. Below we describe the models E, U, D, and then define the predictive function.

4.2 User Model

The level of experience of the user is represented by model E. The following information is stored in this model.

$$E = \{ ED, EA, ES \}$$

where,

- ED - The user's prior knowledge of the domain (i.e. subject or topics described in the hypertext document).
- EA - The user's prior knowledge of the hypertext application.

- ES - The user's prior knowledge of the window-based interfaces and operating systems.

Each of these items has a value between 0 and 1, where 1 is considered an individual with all encompassing knowledge on that topic, and 0 is no previous knowledge. We determine the level of expertise of a user from a questionaire that tests a user's knowledge of the domain, hypertext application, and operating system.

4.3 User Interface Model

The user interface is represented using model U. The following information is stored in this model.

$$U = \{ US, UP, UL \}$$

where,

- US - the screen size.
- UP - The dialog sequence for paging up or down a screen of text.
- UL - The dialog sequence for selecting a link.

The screen size US can be used to compute the pointing time using Fitts' Law [3, 13]. The dialog sequence UP refers to the physical actions such as keystrokes and\or mouse movements needed to go to previous or following pages within a node. The dialog sequence UL refers to the physical actions, such as double-clicking a mouse button, that are needed to select a link. The dialog sequences UP and UL are represented using the keystroke level model [5]. The time to select a particular link in a given document, without accounting for mental decision making, can be computed using the screen size US, the dialog sequences UP and UL, time for the system to respond to the user's input, R (a system dependent value), and the location of the link in the hypertext node. Below we describe the document model and the representation of link location.

4.4 Document Model

The hypertext document is represented using model D. The following information is stored in this model.

$$D = \{ DS, DN, DL, DH, DR \}$$

where,

- DS - The structure of the document (links)
- DN - The total number of nodes in the document
- DL - The total number of links
- DH - The total number of hierarchical links
- DR - The total number of referential links

We distinguish between two types of links: hierarchical and referential links. The semantic definition of a hierarchical link is that the destination node describes a topic that is more specific than the source node (also known as the part-whole relation [29]). For example, the relation between a source node and a destination node that are connected via a hierarchical link is analogous to the relation between section and sub-section in linear text documents such as articles and books. Referential links are defined as any other non-hierarchical link between two nodes and may represent a variety of semantic interpretations such as "see related information", "see also ...",, "see an example of ...", "see definition ...". DeRose [8] expands on the diverse semantic interpretations of hypertext links.

The document structure is represented by DS, where N_i is a node in the document and DN is the total number of nodes in the hypertext document.

$$DS = \{ N_1, N_2, N_3, ..., N_{DN} \}$$

The start node (eg. title page or table of contents) is defined as N_1 and all other nodes are arbitrarily assigned higher numbers. The following information is stored for each node.

$$N_i = \{ NL_i, H_i, R_i, W_i, L_i \}$$

where,

- NL_i - The number of links in node N_i
- H_i - The number of hierarchical links in node N_i
- R_i - The number of referential links in node N_i
- W_i - The number of words in node N_i
- L_i - The links in node N_i

A node N_i may contain one or more links which are represented in L_L

$$L_i = \{ L_{i,1}, L_{i,2}, L_{i,3}, \ldots, L_{i,NLi} \}$$
$$L_{i,j} = \{ N_{i,j}, T_{i,j}, S_{i,j} \}$$

where,

- $N_{i,j}$ - The destination node of the jth link in node N_i
- $T_{i,j}$ - The type of the jth link in node N_i (hierarchical or referential)-
- $S_{i,j}$ - The location of the jth link in the text of node N_i

The location of a link $L_{i,j}$ on the screen can be computed using the location of the link in the text of the node $S_{i,j}$, and the screen size US.

A constraint must be defined that states that in no path which is comprised entirely of hierarchical links can there be a node that appears more than once. This constraint achieves two objectives: 1) it maintains the transitivity of the part-whole relationship [29] where each subsequent node in a hierarchical path is more specific than the previous node, and 2) it prevents circularities for hypertext searches involving only hierarchical links. However, circularities can occur if a user selects one or more referential links in a search path.

Advanced users who understand and can visualize both hierarchical and referential links are less likely to get lost in hyperspace as has been observed in [2, 10, 26]. Some users will follow only hierarchical links to restrict the information they seek, knowing that a referential link can lead them in a circular path. When a hierarchical path is not focusing on the desired information, advanced users are able to select referential links to jump to another, more relevant, section of the document without having to re-trace long hierarchical paths.

4.5 Predictive Function

The function **predict(N_t,D,E,U,Path)** predicts the time it takes a user with expertise E to find a target node N_t in document D using a hypertext interface U along a given Path. The predictive function is defined in Figure 4 and it computes an estimate of the time that a user takes to perform the following tasks:

1. Reach for the Mouse.
2. Read and interpret a Node (Step 3 and 4.6 in Figure 2).
3. Mentally decide on a Link (Step 4.4 in Figure 2).
4. Physically select a Link (Step 4.5 in Figure 2).

The **predict** function is defined as the sum of two components, **H(ES)** and **jump(N_L,D,E,U)**. The function **H(ES)** computes the time it takes a user to reach for the mouse or keyboard and it depends on the user's experience with computer systems. The function **jump()** computes the time it takes a user to go from one node to another.

We define **Path** to be a set of nodes in a path from the first node, N_1, to the target node, N_t. The path can be determined from the model of the document **D**. For each node in the path, the function **jump(N_L,D,E,U)** is used compute the time to read the node, **RN**, mentally decide on a link, **ML**, physically select a link, **PL**, and the time for the system to respond to the user's input, **R**.

The time it takes a user to read a node, **RN**, is based on the number of words in the node, W_L and the time that it takes a user to read a work from the screen, **RW**. The function **RW** depends on the user's knowledge of the domain, **ED**, and the previous experience with hypertext systems, **EA**. For example, an expert user who is familiar with the topics described in the hypertext document and has experience with reading text from computer screens is likely to skim over the text in each hypertext node. Thus, **RW** has a low value for experts.

The time a user will take to decide on a link, **ML**, is directly proportional to the number and type of links in the node, the user's knowledge of the domain, **ED**, and the user's knowledge of hypertext applications, **EA**. The current model assumes that it takes an equal amount of time, **M(ED,EA)**, to decide on hierarchical links, H_L and referential links R_L. However, future research may show that different types of links take a different amount of time to consider and select.

The time it takes to physically select a link, **PL**. depends on the user experience with window-based interfaces, **ES**, the hypertext application, **EA**, the

$$\text{predict}(N_t, D, E, U, Path) = H(ES) + \sum_{N_i \in Path}^{N_t} \text{jump}(N_i, D, E, U)$$

$$\text{jump}(N_i, D, E, U) = RN(W_i, ED, EA) + ML(ED, EA, H_i, R_i) + PL(EA, ES, U, S_{i,j}) + R$$

$$RN(W_i, ED, EA) = W_i \times RW(ED, EA)$$

$$ML(ED, EA, H_i, R_i) = (H_i + R_i) \times M(ED, EA)$$

$$PL(EA, ES, U, S_{i,j}) = \begin{cases} NP(US, S_{i,j}) \times K(ES) + P(EA, ES) + K(ES), & \textit{mouse select} \\ NP(US, S_{i,j}) \times K(ES) + NK(N_j) \times K(ES), & \textit{keyboard select} \end{cases}$$

NK *is system dependent*

NP *is system dependent*

Figure 4: Predictive Functions

user interface design, U, and the location of the link in the hypertext node, $S_{i,j}$. PL can be computed in terms of the functions NP,NK,P, and K, which are described below.

The function NP computes the number of paging commands that a user needs to enter in order to arrive at the screen with the desired link. This can be computed from the screen size US and the location of the link in the hypertext node $S_{i,j}$. The function NP is system dependent and it depends on factors such as font size, work wrap around, the size of embedded graphics. K depends on the user experience with computer systems ES.

For hypertext systems that allow for links to be selected with a mouse, the function P computes the time to point to an area of the screen that displays the desired link. The function P depends on the user's prior experience with the hypertext system EA, and with pointing applications ES. K is the time it takes to press the mouse in order to select the link.

For hypertext systems that allow for links to be selected only from a keyboard, the function NK determines the number of letters that need to be entered to select a link. For example, in the GNU Info hypertext system, links are selected by entering the first few letters of the link name, such that the letters can be used to distinguish the link from other links in the node. The function NK is system dependent.

Figure 5 shows the associated times for RW, H, K, P, R, M, which are partly based on [5, 15], and partly based on our analysis of empirical data collected to verify this model. Our work is partly based on the Keystroke Level Model [5] which is a model that is comprised of keystroke operators for modelling expert users only. In our model, each operator has an associated time that depends on the knowledge that the user has of the domain, the hypertext application, and the computing system. We have also added the RW operator for the time needed to read a word from a hypertext node.

4.6 Cost Function

The predictive functions can be used to compute the quality of a document with respect to minimum predicted information retrieval time. The information retrieval cost function for a given hypertext document is given in Figure 6. We define SPath to be the set of all nodes in the shortest path from N_1 to N_t. We define Cost to be the sum of the time to find each item in the document. Thus, alternative designs can be compared in terms of information retrieval cost.

$$RW(ED, EA) = \begin{cases} 0.05, & ED + EA > 1.25 \\ 1.1, & ED + EA < 1.25 \end{cases}$$

$$H(ES) = \begin{cases} 0.4, & ES > 0.75 \\ 1.2, & ES < 0.75 \end{cases}$$

$$K(ES) = \begin{cases} 0.08, & ES > 0.75 \\ 0.5, & ES < 0.75 \end{cases}$$

$$P(EA, ES) = \begin{cases} 0.8, & EA + ES > 1.5 \\ 1.1, & EA + ES < 1.5 \end{cases}$$

$$M(ED, EA) = \begin{cases} 0.1, & ED + EA > 1.2 \\ 0.4, & ED + EA < 1.2 \end{cases}$$

R *is system dependent*

Figure 5: Times for Keystroke Level Operators

Some items in a document may be more frequently referred to by a given set of users. Where the set of frequencies for each item in the hypertext document is known, a weighted cost function can be defined for information retrieval. The relative frequency is the relative number of times that an item is searched by a typical set of users. This cost may be a more accurate reflection of the quality of a document where some items are more frequently referred to than others.

5 EMPIRICAL OBSERVATIONS AND ANALYSIS

The predictive model was used to evaluate the effectiveness of an existing hypertext system called HRS [27]. HRS provides overview maps, keyword searching and semantic based searching of information [27, 16]. HRS is currently available on an IBM PC network at the University of Waterloo that serves some of the computing needs of 2000 undergraduate students. The objective of the evaluation focused on a collection of hypertext documents that describe programming languages such as C and Fortran. Some empirical observations are described below which reveal the strengths and limitations of our model.

A preliminary experiment was conducted to collect data for deriving parameters of the predictive model. A hypertext document on the C programming language was used. A set of 8 retrieval tasks was given to 16 test subjects. Thus this experiment explores the ability of users to retrace paths and previously viewed nodes, and their ability to retain a model of the organization in their minds.

It was found that on a first search, over two thirds of users took over 150 seconds to find the answer. However on the second search, over two thirds of users were able to complete their searches in under 50 seconds. All users at some point were able to complete one or more searches in under 25 seconds. This result suggests that users can retain a model of the local organization of a document and improve their searches to near expert speed, regardless of their previous experience. The observations also suggest that the user's mental model changes quickly as is reflected by the improving search times.

The predicted time for the first search task (how to open a file in the C programming language) was 22.77 for experts (ES > 0.75, EA+ES > 1.5, ED+ES > 1.5) and 78.1 for novices. The answers to the search tasks were located at a similar depth and each was assigned a similar predicted time. The predicted

Cost Function:

$$\text{Cost} = \sum_{i=1}^{DN} \text{predict}(N_i, E, U, D, \text{SPath})$$

$$\text{Weighted Cost} = \sum_{i=1}^{DN} \text{relative\_frequency}(N_i) \times \text{predict}(N_i, E, U, D,)$$

Constraints:

1) $$0 \leq \text{relative\_frequency}(N_i) \leq 1 \text{, for all } 0 < i < DN$$

2) $$\sum_{i=1}^{DN} \text{relative\_frequency}(N_i) = 1$$

Figure 6: Information Retrieval Cost Function for Hypertext Documents

times are within the range of our empirical observations. A larger sample size is needed to compute the reliability of our model. It should also be noted that in two cases, users were able to complete these tasks in 7 seconds. This suggests that in some cases little or no cognitive decision making occurs when users retrace previously viewed paths.

A second experiment was conducted with 12 subjects and a more detailed questionaire. The questionaire was used to assign a value between 0 and 1 for each user in terms of domain knowledge ED, hypertext application knowledge, EA, and knowledge of window-based interfaces, ES. The same document and nine questions from the previous experiment were used. Each user's actions were recorded and statistically analyzed.

A correlation analysis was done between each user's knowledge and the total search time, number of nodes visited, and percentage of correct answers found. Correlation can be used to determine if there is consistent evidence between the type of knowledge that a user has, and the observed search behaviour. A correlation value of +1 indicates a perfectly positively correlated relationship and a value of -1 indicates a perfectly negatively correlated relationship. The sample correlation co-efficient and pearson's product moment co-efficient were used. From the significance tables for pearson's product moment co-efficient, it can be determined that for a sample size of 12 observations with a significance level of 0.05, pearson's correlation co-efficient must exceed 0.50. A significance level of 0.05 means that there is less than a 5% chance that the correlation is the result of pure chance.

The correlation analysis showed that there is a statistically significant correlation between percent of correct answers found by a user and the sum of the user's domain knowledge and application knowledge (pearson's correlation co-efficient = 0.56, sample correlation co-efficient = 0.51). This is consistent with our belief that the mental operator, ML, time to decide on a link, should be associated with the user's domain knowledge, ED, and the user's knowledge of hypertext applications, EA. There is also a statistically significant correlation between the user's knowledge of hypertext applications and knowledge of window-based systems and the number of nodes visited (pearson's correlation co-efficient = 0.52, sample correlation co-efficient is 0.47). This is consistent with our belief that the operator PL, time for physically selecting a link, should be associated with the user's knowledge of hypertext applications, EA, and the user's knowledge of window-based systems, ES. The correlations between user's knowledge and total search times were not statistically significant and can not reliably used to make any statistical inference. A larger sample size is needed to determine a statistically significant correlation.

A t-test was also used to determine if expert users found information significantly faster than novices. The computed t-value can then be compared to t-distribution tables to determine the probability that the difference in observations is the result of pure chance. Since there were 7 novice users, and 5 expert users, the t(5) distribution table was used for comparison. For a 0.05 level of significance, the t-value needs to exceed 2.571.

The significance levels that were computed show that only in two cases did expert users find information significantly faster than novices. In question 3 (t=3.25), and question 7 (2.49), experts were significantly faster. Both of these questions required a user to find an answer to a question which was part of a previous question. Questions 3 and 7 were worded differently but asked the same question. This result suggests that experts are significantly better at retracing a previously viewed hypertext path than novice users.

A matched t-test was also used to see if experts found information faster than novices. This test compares the difference (d) in mean time for each question, and uses this difference to compute the t-value. The t-value, computed for novices and experts in each of the 9 questions, was found to be 0.94 and it is not statistically significant. This suggests that, although initially novices are slower than experts at finding information, novices can quickly improve their searches. It can be concluded that over the nine questions tested, experts were not significantly faster than novices.

Using the KLM-H Operators values defined in this paper, the root mean squared error was found to be 31%. A closer examination of the results show that in certain cases, users had a pre-made decision as to their hypertext path and link selection. Hence, in certain nodes, the total time spent at the node was very small. In other cases, a user would spend a significantly higher amount of time, deciding on a link, than what was predicted. In these cases, the user had read the information and perhaps did not understand the information and found it difficult to decide on a link. These effects, in many cases, cancelled each other since in one case the predicted time is under-estimated, and in the other case the predicted time is over-estimated. Future work could be done to modify the mental operator depending on the user's current search strategy.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a framework for modelling the cognitive search tasks involved in hypertext searching. We have proposed a model for the user, the document, and the interface. The information that needs to be contained in these models has been defined and we have described the interaction of these models.

A predictive model has been presented that allows for estimating retrieval times from hypertext documents based on a model of the document, user expertise, and user interface. The predictive model can also be used to evaluate the quality of a hypertext document in terms of retrieval time.

The model's predictions have been found to be in the range of empirical observations. Observations have also shown that hypertext users can improve their searching abilities very quickly with successive searches, suggesting that the user model changes quickly during a searching session. We also observed that, in some cases, such as retracing paths, little cognitive decision making is involved.

Future extensions could account for diverse search strategies and path selection of users. This could be done based on large sets of data collected from empirical observations that reveal search strategies given a state of the user model. Future work could also include:

1. The elapsed time after which users are likely to abandon a search in frustration.
2. The elapsed time after which users are likely to abandon a search because they believe the information is not in the document.
3. The initial time spent interpreting the search task requirements and developing a goal, subgoals, and search strategies.

4. The quantitative effect of overview maps on search time.
5. The effect of the user's learning ability on decision making time.
6. The effect of multimedia on a user's ability to assimilate information.
7. The effect of slow response times on a user's frustration level and search strategy.

While a great deal remains to be learned from cognitive search behaviour in hypertext searching, preliminary models can be designed based on existing empirical observations. Future work will require more empirical data to be collected in controlled settings so as to try isolate and quantify the effect of particular hypertext designs on searching. The challenge will be to correlate user characteristics with search behaviour and the use of statistical inference to determine casual relationships.

References

[1] Bush, Vannevar. 1945. " As We May Think", Atlantic Monthly, No. 176 (July), pp. 101-108. Reprinted in CD ROM: The New Papyrus, edited by Steve Lambert and Suzanne Ropiequet, Redmond, WA: Microsoft Press, 1986

[2] Campagnoni, F.R. and Kate Ehrlich. "Information Retrieval Using a Hypertext-Based Help System", Proceedings of the ACM Conference on Information Retrieval (SIGIR '89), pp. 212-220

[3] Card, S.K., English, W.K., Burr, B.J.. 1978. "Evaluation of Mouse, Rate Controlled Isometric Joystick, Step Keys and Text Keys for Text Selection on a CRT", Ergonomics, Vol. 21, No. 6 pp. 601-613

[4] Card, Stuart K., Thomas P. Moran and Allen Newell. 1980. "The Keystroke Level Model for User Performance Time with Interactive Systems", Communications of the ACM, Vol. 23, No. 7, pp. 396-410.

[5] Card, Stuart K., T.P. Moran, and Allen Newell. 1983. The Psychology of Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Erlbaum.

[6] Carmel, Erran, Stephen Crawford, and Hsinchu Chen. 1992. "Browsing in Hypertext: A Cognitive Study", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, No. 5 (September/October), pp. 865-884

[7] Conklin, Jeff. 1987. "Hypertext: An Introduction and Survey", Computer, Vol 20, No. 9 (September), pp. 17-41.

[8] DeRose, S.J. 1989. "Expanding the notions of links", Proceedings of the ACM Conference on Hypertext (Hypertext '89), November 5-8, Pittsburgh, Pennsylvania, pp. 249-257

[9] Egan, Dennis E., Joel R. Remde, Louis M. Gomez, Thomas K. Landauer, Jennifer Eberhardt, and Carol C. Lochbaum. 1989. "Formative Design-Evaluation of the SuperBook", ACM Transactions on Information Systems, Vol. 7, No. 1 (January), pp. 30-57.

[10] Egan, Dennis E., Joel R. Remde, Thomas K. Landauer, Carol C. Lochbaum, and Louis M. Gomez. 1989. "Acquiring Information in Books and SuperBooks", Machine-Mediated Learning, Vol. 3, pp. 329-277

[11] Egan, Dennis E., Joel R. Remde, Thomas K. Landauer, Carol C. Lochbaum, and Louis M. Gomez. 1989. "Behavioral Evaluation and Analysis of a Hypertext Browser", Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'89), pp. 205-210.

[12] Frisse, Mark E. 1988. "Searching for Information in a Hypertext Medical Handbook" Communications of the ACM, Vol. 31, No. 7, pp. 880-886.

[13] Gillan, Douglas J., Kritina Holden, Susan Adam, Marianne Rudisill, and Laura Magee. 1990. "How does Fitts' Law Fit Pointing and Dragging?", Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'90), pp. 227-242.

[14] Hardman, L. 1989. "Evaluating the usability of Glasgow Online hypertext", Hypermedia, Vol. 1, No. 1, pp. 34-63.

[15] John, B.E. 1990. Extensions of GOMS analysis to expert performance requiring perception of dynamic visual and auditory information", Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'90), pp. 107-115.

[16] Kamel, Mohamed, and Yuri Quintana. 1990. "A Graph-Based Knowledge Retrieval System", Proceedings of the IEEE Conference on Systems, Man, Cybernetics, Los Angeles, California, November 4-7th, 1990.

[17] Marchionini, Gary. 1987, "An invitation to browse: designing full text systems for novice users", Canadian Journal of Information Science, Vol. 12, no. 3, pp. 69-79.

[18] Marchionini, Gary, and Ben Shneiderman. 1988. "Finding Facts vs. Browsing Knowledge in Hypertext Systems", Computer, Vol. 21, No. 1 (January), pp. 70-80.

[19] Marchionini, Gary. 1989. "Information-seeking strategies for novices using a full text electronic encyclopedia", Journal of American Society for Information Science, Vol. 40, No. 1, pp. 54-66.

[20] Monk, Andrew F., Paul Walsh, and Alan J. Dix. 1988. "A Comparison of Hypertext, Scrolling and Folding as Mechanisms for Program Browsing", Proceedings of the British Conference on Human-Computer Interaction (HCI'88), In People and Computers IV, Cambridge, U.K.: Cambridge University Press, pp. 421-435.

[21] Nelson, Theodore H. 1981. Literary Machines. Published by author. Subsequent editions in 1982, 1983, 1984, 1987, 1990.

[22] Nielsen, Jakob. 1987. "Using Scenarios to develop user friendly videotext systems", Prodeedings of NordDATA '87 Joint Scandinavian Computer Conference, Trondheim, Norway, June 15-18, pp. 133-138.

[23] Nielsen, Jakob, and Uffe Lyngbaek. 1989. "Two Field Studies of Hypermedia Usability", Proceedings of the UK Conference on Hypertext, University of York, U.K. June 29-30, In HYPERTEXT II: State of the Art, edited by Ray McAleese and Catherine Green, Oxford, England: Intellect; Norwood, N.J. Ablex Pub. Corp. pp. 64-72.

[24] Nielsen, Jakob. 1989. "The matters that really matter for hypertext usability". Proceedings of the ACM Conference on Hypertext (Hypertext '89), November 5-8, Pittsburgh, Pennsylvania, pp. 239-248.

[25] Nielsen, Jakob. 1990. Hypertext and Hypermedia. San Diego, CA: Academic Press.

[26] Nielsen, Jakob. 1990. "The art of navigating through hypertext", Communications of the ACM, Vol. 33, No. 3, pp. 298-310.

[27] Quintana, Yuri, Mohammed Kamel, Andrew Lo. 1992. "Graph-based Retrieval of Information in Hypertext Documents". Proceedings of the ACM Conference on Systems Documentation (SIGDOC'92), Ottawa, Ontario, October 13-16, pp. 157-168.

[28] Rouet, Jean-Francois. 1992. "Cognitive Processing of Hyperdocuments: When Does Nonlinearity Help?", Proceedings of the ACM European Conference of Hypertext (ECHT'92), pp. 131-140.

[29] Winston, Morton E., Roger Chaffin, and Douglas Hermann. 1987. "A Taxonomy of Part-Whole Relations", Cognitive Science, Vol. 11, pp. 417-444.