

Proposed Evaluation Framework for Adaptive Hypermedia Systems

Arpita Gupta, P. S. Grover

Department of Computer Science, University of Delhi,
Delhi – 110 007, India
arpita_aggarwal@hotmail.com, psgdelhi@yahoo.com

Abstract. Although a number of frameworks exist for the evaluation of Adaptive Hypermedia Systems (AHS), recently suggested layered frameworks have proved useful in identifying the exact cause of the adaptation failure or any other error in the system. This paper presents an evaluation framework for AHS for internet which is an extension of the layered frameworks and adds new dimensions to them. It treats evaluation as an integral part of development process of AHS and also evaluates the successful access of AHS on the internet. The framework has four dimensions which are orthogonal to each other – Environment – the environment in which AHS is accessed, Adaptation – the type of adaptation used, Development Process – software engineering life cycle steps used for developing AHS and the Evaluation Modules – the layers of AHS which are evaluated in context of other dimensions.

1 Introduction

Adaptive Hypermedia Systems (AHS) are designed and built with the intention of providing tailor made information to individual users according to their preferences, goals and knowledge. With the advent of internet as a common source of information, they have found a platform to reach heterogeneous groups of users using different devices for assessing AHS. With this increases the challenge of catering to a wide variety of users in differing environments and also the added responsibility of working without making mistakes since a single mistake can make the user lose trust in the system – maybe forever. Therefore, evaluating the AHS is of utmost importance. Moreover, it is equally important to have a correct method of evaluation since an incorrect method can lead to wrong conclusions [3].

Earlier evaluation studies compared adaptive versions of the system with the non-adaptive versions [2, 4]. A major criticism of this approach was that the non-adaptive versions – usually implemented using adaptive version with their adaptivity switched off – were not “optimal” [11].

Recently some layered evaluation frameworks were suggested which do not treat evaluation as a “monolithic” process but instead divide it into layers [3, 20, 23]. This

approach helps in identifying the exact cause of the adaptation failure or any other error. These evaluation frameworks basically differ in layer granularity and do not take “extensibility or maintainability” of the AHS into consideration [10]. Another evaluation framework – Extended Abstract Categorization Map (E-ACM) [21] has been suggested to guide adaptation evaluation and design. Most of them do not take the development process into consideration resulting in detection of errors and weaknesses in the system which can prove to be expensive to correct at a later stage. Moreover issues like maintenance of the system, the environment in which they will be used – location and devices accessing AHS, etc., have not been addressed in the current frameworks.

There can be a number of factors which affect the evaluation process of an AHS. Internet provides the opportunity of accessing AHS using a variety of devices like desktops, mobiles, PDAs, in any location of the world, to the users having diverse skills, capabilities and knowledge. The AHS itself can belong to any application domain having some specific characteristics. All these elements form the environment of the AHS. The adaptation in the AHS can be static or dynamic depending on the time and process of adaptation.

Similar to software engineering, AHS also involves the analysis, design, implementation and maintenance phases of development process. During these phases, they should be evaluated for the validity of input acquisition, correctness of inferences drawn from these inputs, correctness of various models created by the AHS, the adaptation decisions taken based on these models and their final presentation to the user.

We propose an evaluation framework treating the evaluation as an integral part of the development process of AHS and taking the accessing environment and the type of adaptation provided by AHS into consideration while evaluating individual modules of AHS. The framework consists of four orthogonal dimensions: Environment, Adaptation, Development process and the Evaluation modules. Next Section describes the framework.

2 Proposed Evaluation Framework

The proposed framework is an extension of layered evaluation frameworks where the layers need to be evaluated in context with other dimensions. The benefits of the framework are: (i) it allows a structured, layered view to better understand the various aspects of AHS (ii) it can be used as a conceptual framework for evaluating existing approaches for AHS (iii) it may be used during the development of next generation AHS using software engineering steps.

The framework consists of 4 dimensions - Environment, Adaptation, Development Process, and Evaluation Modules. These dimensions are orthogonal to each other i.e. all the evaluation modules should address all the components of environment and adaptation during each phase of development process.

Figure 1 shows our proposed evaluation framework and the following sections describe these dimensions and their components.

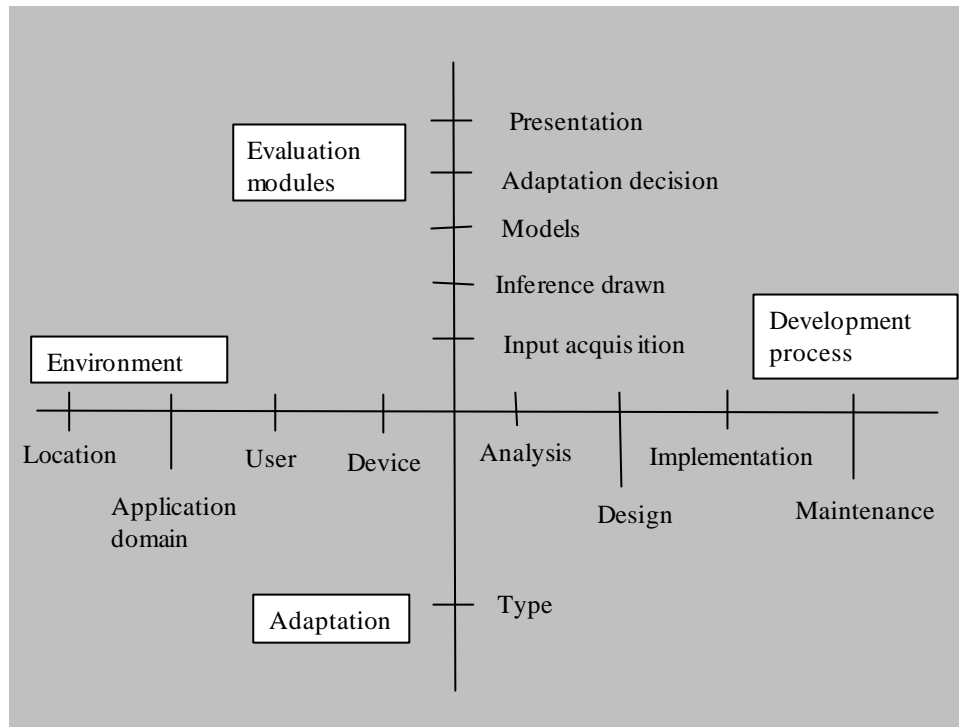


Figure 1. Proposed Evaluation Framework for AHS

2.1 Environment

The first dimension – environment – is the set of conditions to which AHS has to adapt itself i.e. they are the circumstances of consumption for AHS. There can be innumerable variables affecting the environment; we group them under the following components:

2.1.1 Device: Advent of web-capable appliances with limited abilities such as PDAs and mobile telephones along with desktops – have made one-size-fit-all paradigm obsolete since the range of hardware and software used at client side of web-based systems is extremely wide. AHS need to be evaluated for the correct acquisition of the device characteristics and smooth running with hardware features like display sizes, local storage size, method of input, processing speed and software features like browser versions, available plug-ins, Java and JavaScript etc. Kobsa et al [16] have discussed acquisition of such data by the AHS.

2.1.2 User: The personal characteristics of user such as demographic data, user knowledge, user skills and capabilities, user interests and preferences, his goals and plans have been used by many AHS for adaptation [2, 22]. Along with these features,

AHS should be evaluated for adaptation according to all users including disabled and elderly users [15].

2.1.3 Application Domain: AHS can be developed for a wide range of applications which differ in characteristics. Brusilovsky [5] have identified several application domains and existing AHS for them. The application specific characteristics constitute critical parameters while evaluating AHS and should be evaluated along with general characteristics of AHS since the traditional concerns change with the application domain.

2.1.4 Location: Information about the geographical location of the accessing device can be used to filter and adapt or recommend the content of AHS and should be evaluated for the correct delivery of the same. Kobsa et al [16] have given methods for acquisition of information about the location and their use in adaptation procedure.

2.2 Adaptation

The second dimension of the framework is adaptation which can be of two types: **static** adaptation and **dynamic** adaptation, depending upon the time and process of adaptation. Static adaptations are specified by the author at the design time or determined once only at the startup of the application. Fink et al [9] used static adaptation in AVANTI project. Dynamic adaptation occurs during runtime depending on various factors like inputs given by the users during use, changes in user model, adaptation decision taken by AHS etc. Kappel et al [13] have distinguished three options for dynamic adaptation – immediate dynamic adaptation i.e. adaptation done as soon as context changes, deferred dynamic adaptation i.e. adaptation done only after the user has requested the page which is subject to adaptation, and periodic adaptation i.e. adaptation is done periodically. Example of system having dynamic adaptation is AHA [8].

2.3 Evaluation Modules

The third dimension of evaluation framework consists of evaluation modules which need to be considered for evaluation of AHS. These modules have been suggested in the layered frameworks [3, 20, 23]. Our perspective is to evaluate them with respect to other dimensions of the framework.

2.3.1 Input Acquisition: Inputs are required from the environment as well as from the user. These can be taken manually (i.e. user feeds them), automatically (i.e. system takes the input itself e.g. Type of device, its screen size, location etc) or semi automatically (i.e. combination of both – some input through user, some automatically) [13].

The inputs taken by the system – either manually or automatically might not carry any semantic information, but they need to be evaluated for the reliability, accuracy, precision, latency, sampling rate, so that the inferences drawn from them are valuable.

This needs to be done at all stages of development process – analysis, design, implementation and maintenance phases, for both static and dynamic adaptations, for all intended devices, users, locations and application domains.

2.3.2 Inferences Drawn: Previous layer was involved with the data collection, this layer gives “meaning” or “semantics” to it i.e. it draws inferences from it. Evaluators need to check if these inferences or the conclusions drawn by the system concerning the user-computer interaction are correct since it is not necessary that there will be a direct – one to one mapping between raw data and their semantically meaningful counterparts.

Moreover, inputs given by various users of different devices or application domain might need different interpretations. Evaluators need to check if all such interpretations have been analyzed, designed and implemented in the AHS for both static and dynamic adaptations.

2.3.3 Models: For achieving the required adaptations, various models are created by the system. Benyon and Murray [1] specified three models – user model, domain model, interaction model. Nora Koch [18] has described four models for carrying out the adaptation – user model, navigation model, presentation model, and adaptation model.

These models are based on the inferences drawn in the previous stage and are supposed to imitate the real world. They need to be evaluated for validity i.e. correct representation of the entity being modeled, comprehensiveness of model, redundancy of model, precision of the model, sensitivity of the modeling process [20].

2.3.4 Adaptation Decision: Given a set of properties in the user model, sometimes there can be more than one adaptation possible. In this module, evaluation of the most “optimal” adaptation is done using criteria like necessity of adaptation, appropriateness of adaptation, acceptance of adaptation [20]. Careful evaluation is needed to ascertain that increase in adaptation is not resulting in decreased usability [5].

2.3.5 Presentation: This module involves the human-computer interaction and needs to be evaluated for criteria like completeness of the presentation, coherence of presentation, timeliness of adaptation, user control over adaptation [20].

2.4 Development Process

The fourth dimension of the framework is the development process comprising of phases of software life cycle i.e. analysis, design, implementation and maintenance. Benyon and Murray [1] gave a star approach to interactive system development taking evaluation as central element and system analysis, specification of user requirements, design, prototype and implementation as the peripheral elements.

During each phase of this dimension, evaluation of individual elements of environment, adaptation and evaluation modules is done with respect to each other.

2.4.1 Analysis: This phase involves gathering information about the problems of current system, and/or identifying the requirements and constraints of the system to be developed. Main components of this phase are:

- *Functional analysis:* establishes the main functions that the system is expected to perform and how it is to perform.
- *Environment analysis:* This analyzes the environment in which the system is expected to be accessed – including the physical aspects like location, device and other aspects like application domain, type of user.
- *User and task analysis:* This determines the scope of cognitive characteristics like user's preferences, goals, knowledge and other attributes required in user model e.g. search strategy required, assumed mental model etc.
- *Interface analysis:* It identifies features like effectiveness, learnability, flexibility and attitude required of the system.
- *Data analysis:* This involves the analysis of input acquisition to identifying the data to be stored and manipulated by the system, and to understand and represent the meaning and structure of data in the AHS.
- *Analysis of Models:* This involves the analysis of various models maintained by the system such as user model, domain model, navigation model, adaptation model.

To evaluate this phase, checklists can be prepared for different types of analysis mentioned above, corresponding to various components of different dimensions of the evaluation framework. For example, a checklist is prepared for the desktop computer's functional requirements of static adaptation for input acquisition; another checklist for the PDAs for the same specifications is prepared.

2.4.2 Design: Design phase defines the overall system organization by transforming the functions and tasks defined during analysis phase into software components and their externally visible properties of those components and their relationships. It is recommended to design the adaptive parts of the system in parallel with the whole system so as to have a successful adaptation [11]. Some of the components for this phase are:

- *Architectural Design:* Many architectural designs have been suggested for the adaptive systems [1, 12, 19]. The modular architecture model presented in figure 2 is especially designed for our evaluation purpose.

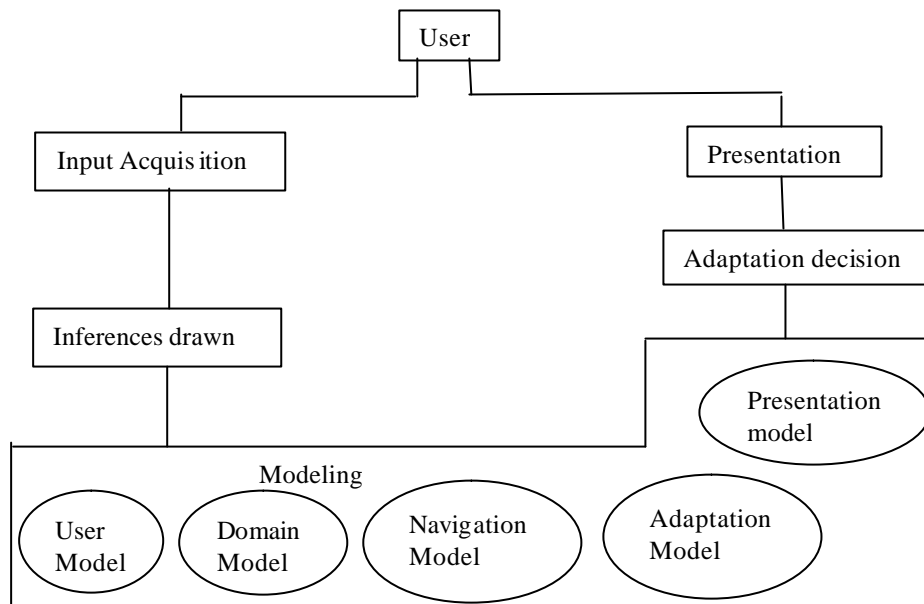


Figure 2. Architecture model for evaluation of adaptive hypermedia system

In this model, “Input acquisition” module acquires input by the user through keyboard or mouse or takes it from the environmental factors such as the device and location accessing the AHS. This input is verified and then passed on to the next module “Inferences drawn” for drawing logical inferences and conclusions which are then stored in some dynamic models created by the system such as user model. The “modeling” module consists of various models – both static and dynamic such as user model, domain model, navigation model, presentation model and adaptation model. Different AHS can have one or more such models which are used at various levels of adaptation. Depending on some these models, “adaptation decision” module takes the decision so as to choose the best adaptation techniques out of the available ones in the system. Then the “presentation” module presents the final content and links to the user.

Evaluation of architectural design is important because it defines constraints on implementation and maintenance, making it easier to reason about and manage changes, and defines system’s quality attributes.

- *Content Design:* Content forms one of the most important aspect of the AHS since it is for the content that the AHS exists. Therefore, it should be verified for accuracy and precision. Moreover, there is a need to identify and evalu-

ate the content which would be visible to different users (according to their individual user model), on different devices and location.

- *Navigation Design:* Since a number of navigational techniques are present such as direct guidance, link hiding etc [5] which can be used with various navigation aids like icons, graphics, there is a need to ascertain that the chosen one is appropriate for content and consistent with the heuristics leading to high quality interface designs and also ensure that aids like site maps and table of contents are designed into the system.

Evaluation of navigational design is needed to ensure the maintenance of coherence of the overall system as the user moves across application since it is related to AHS's underlying communicative performance.

- *Interface Design:* Since user interface is the "first impression" of the system, a poorly designed interface might disappoint user regardless of the value of content, sophistication of architectural design and navigation techniques used. Therefore, careful evaluation is necessary for well structured ness and ergonomically sound interface designs.

Design phase can be evaluated by using metrics such as structural complexity metrics, navigational metrics, usability metrics etc. Moreover, a good architecture exhibits low coupling and high cohesion in terms of some decomposition of functionality [14].

2.4.3 Implementation: During the actual implementation of the system, evaluation of each module of AHS should be carried out individually and then in integration with the other modules for the successful adaptation at various levels – both static and dynamic, for different users, on different devices. Some metrics like behavioral complexity, reliability metrics, precision, software size and length metrics help in evaluation of the system as a whole. Moreover, evaluation of tools can be done in relation to the activities in which they are used to indicate their availability for each kind of activity and how the tool supports it.

2.4.4 Maintenance: AHS might require updation at any moment; therefore, during the design phase care should be taken to design hyperspace in modular way so that change in structure can be made by changing the relations among these modules. Automatic link generation should be preferred over static links to preserve the link coherence. Maintenance of static links is very complex as any change in a node position in hyperspace necessitates it to revise all the documents that include links to this node, in order to update them.

Content maintenance can be made easy by storing contents apart from the concept structure or the navigational options since contents are external and easily updateable. Finally, database should be used to store information about different system components to facilitate the management and maintenance of these components and guarantee data consistency [7]. Checklists can be prepared for these and metrics can be used

to measure the ease of maintenance such as complexity metrics, reuse metrics or expandability metrics.

Table 1 gives an example to illustrate the way in which to use the framework. While developing an adaptive tutoring system for internet, during the implementation phase, the precision of the content is measured for the presentation module and the values are filled in the table 1. The values are filled by using various metrics appropriate for different stages and can be in any unit of measurement. For the complete evaluation, many such tables would be required for various phases.

Table 1. An Example to show how to use the framework dimensions (values are only for demonstration purpose and are not exact).

Precision of content during implementation phase of presentation module		Adaptation		
			Static Adaptation	Dynamic Adaptation
Environment	Device	PDA	85%	80%
		Mobile	80%	80%
		Desktop	95%	96%
	User	Novice	70%	76%
		Average	90%	90%
		Expert	80%	85%
	Location	.		
		.		
		.		
	Application Domain	.		
		.		
		.		

3 Conclusion

Our proposed evaluation framework integrates the AHS development process, the accessing environment, the different types and levels of adaptations involved in AHS and the evaluation modules of layered frameworks. Several factors that impact the AHS evaluation can be organized around these framework perspectives.

The framework is a mechanism to gain insight into and an understanding of AHS for internet. It can be used for summational evaluations once the AHS has been completed by replacing the “development process” with “initial goals and achieved goals” and checking with the rest of the three dimensions. It can also be used for formative evaluations during the development of the system by establishing goals for each phase and then compare the actual results.

The dimensions and their elements suggested in the framework have been addressed more or less globally. Subfactors can be established for each element which can be evaluated objectively or subjectively.

We are in the process of developing and evaluating an adaptive tutoring system with the authoring tool AHA using this framework where analysis phase has checklists and set of goals prepared according to the requirements. Metrics largely are being used during design, implementation and maintenance phases for the purpose of evaluation. The results of this study will be reported later on.

4 References

1. Benyon D., Murray D.: Adaptive Systems: from intelligent tutoring to autonomous agents. *Knowledge-Based Systems*, 6(4), 197-219 (1993)
2. Boyle C., Encarnacion A. O.: Metadoc: An Adaptive Hypertext Reading System. In P. Brusilovsky et al. (Eds.), *Adaptive Hypertext and Hypermedia*, ©1998 Kluwer Academic Publishers, pg 71-89, (1994)
3. Brusilovsky P., Karagiannidis C., Sampson D.: The Benefits of Layered Evaluation of Adaptive Applications and Services. In Weibelzahl S., Chin D. N., and Weber G. (eds), *Empirical evaluation of Adaptive systems*, Proceedings of workshop at the eighth international conference on user modeling, UM2001, pg 1-8, Freiburg
4. Brusilovsky P., Eklund J.: A Study of User Model Based Link Annotation in Educational Hypermedia. *Journal of Universal Computer Science*, vol. 4, no. 4 (1998), pg 429-448, (1998)
5. Brusilovsky P.: Methods and Techniques of Adaptive Hypermedia, P. Brusilovsky et al (eds.), *Adaptive Hypertext and Hypermedia*, 1-43, Kluwer Academic Publishers, 1998, printed in the Netherlands
6. Brusilovsky P.: Efficient Techniques for Adaptive Hypermedia, In C. Nicholas and J. Mayfield (eds): *Intelligent hypertext: Advanced techniques for the world wide web*. LNCS, 1326, Berlin:Springer-Verlag, 12-30
7. Carro, R.M.: Adaptive Hypermedia in Education: New Considerations and Trends. Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (Orlando, Florida), Vol. 2, ISBN: 980-07-8150-1, 452-458, (2002)
8. De Bra, P., Smits D., Stash N.: AHA! The next generation. *ACM Conference on Hypertext and Hypermedia*, May 2002
9. Fink, J., Kobsa, A, Nill, A.: Adaptable and adaptive information provision for all users, including disabled and elderly people. In *The New Review of Hypermedia and Multimedia*, 4, 163-188
10. Gupta A., Grover P. S.: Comparison of Evaluation Frameworks for Adaptive Hypermedia. To be published in *Proc. of 2nd International Conference on Quality, Reliability and Information Technology (ICQRIT Dec 2003)*, New Delhi
11. Höök K.: Steps to take before intelligent user interfaces become real. *Interacting with Computers*, 12, pg 409-426, (2000)
12. Jameson, A.: *Systems That Adapt to Their Users: An Integrative Perspective*. Saarbrücken: Saarland University, (2001)

13. Kappel, B. Pröll, W. Retschitzegger, W. Schwinger: Customisation for Ubiquitous Web Applications - A Comparison of Approaches. *Int. Journal of Web Engineering and Technology (IJWET)*, Volume 1, No. 1, 2003, pp. 79-111, [ISSN 1476-1289]
14. Kazman R., Clements P., Bass, L.: *Software architecture in Practice*. Addison-Wesley, 1998, pg 218
15. Kobsa, A., Stephanidis, C. : Adaptable and Adaptive Information Access for All Users, Including Disabled and Elderly People. *Proceedings of 2nd Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT'98, Pittsburg, USA, June 20-24 (1998)*
16. Kobsa A., Koenemann J., Pohl W.: Personalised Hypermedia Presentation Techniques for Improving Online Customer Relationships. *The Knowledge Engineering Review*, Vol. 16:2, 111-155, (2001), Cambridge University Press
17. Mendes E., Hall W., Harrison R.: Applying Metrics to the evaluation of Educational Hypermedia Applications, *Journal of Universal Computer Science*, vol. 4, no. 4 (1998), pg 382-403
18. Nora Koch, PhD Thesis : *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. Ludwig-Maximilians-University of Munich, Germany, December 2000
19. Oppermann, R.: Adaptively supported adaptability. *International Journal of Human Computer Studies*, 40(3), 455-472, (1994)
20. Paramythis A., Totter A., Stephanidis C.: A Modular Approach to the Evaluation of Adaptive User Interfaces, in Weibelzahl, S. Chin, D.N., Weber, G. (eds), *Empirical Evaluation of Adaptive Systems*, *Proceedings of workshop at the eighth International Conference on User Modeling, UM2001*, pg 9-24, Freiburg, (2001)
21. Tobar, C. M.: Yet Another Evaluation Framework. In: Weibelzahl, S. and Paramythis, A. (eds.). *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems*, held at the 9th International Conference on User Modeling UM2003, Pittsburgh, (2003), pp. 15-24
22. Weber, G, Specht, M.: User modeling and adaptive navigation support in WWW-based tutoring systems.. In a Jameson, C Paris and C Tasso (eds) *User Modeling: Proceedings of the sixth International Conference Springer-Verlag*, (1997), 289-300
23. Weibelzahl S., Lippitsch S., Weber G. (2002): Advantages, Opportunities, and Limits of Empirical Evaluations: Evaluating Adaptive Systems, *Künstliche Intelligenz*, 3/02, 17-20

The First Click is the Deepest: Assessing Information Scent Predictions for a Personalized Search Engine

Karen Church, Mark T. Keane & Barry Smyth

Adaptive Information Cluster, Department of Computer Science,
University College Dublin, Belfield, Dublin 4, Ireland
{karen.church, mark.keane, barry.smyth}@ucd.ie

Abstract. “First-click behavior” describes one of the most commonly occurring tasks on the Web, where a user submits a query to a search engine, examines a list of results and chooses a link to follow. Even though this task is carried out a billion times a day, our understanding of the factors influencing this behavior is poorly developed. In this paper, we empirically evaluate information scent predictions for first-click behavior in the use of a personalized search engine, called I-SPY. Our experiments show that the predictive accuracy of current information foraging approaches is not good. To conclude, we advance a framework designed to understand first-click behavior and guide future research.

1 Introduction

Almost every time someone opens a web browser they carry out a very simple behavioral sequence leading to their first click to some distant website. This sequence involves the user submitting a query to some search engine, scanning a list of returned results and choosing to click on a selected link. Though this, apparently simple, “first-click behavior” is incredibly commonplace, it is still not wholly clear how it should be modeled and what factors influence the behavior. The best predictive models we have of the behavior are information foraging and scent theories of web usage [5, 6, 7, 8, 25, 26]. Hence, in this paper, we report an empirical evaluation of information scent predictions based on an empirical study of web usage in a personalized search engine, called I-SPY [29, 30]. We find that these approaches do not make accurate predictions, prompting us to re-assess the cognitive basis of first-click behavior.

In the next section, we outline information scent approaches and the I-SPY system. Then, we sketch the empirical study of I-SPY. Next, we describe our empirical evaluation of information scent techniques and present the results found when these techniques are applied to the I-SPY data. In part response to the predictive failure found, we advance a general framework for assessing first-click behavior with a view to understanding what might be important in I-SPY. This same framework also helps us understand what it is that information scent approaches are trying to capture and how they may need to be modified to do better in the future.

2 Information Scent & I-SPY

Taxonomic studies have shown that users adopt several distinct types of behavior when using the Web, one of which is the specific goal-driven search for answers to specific questions [3, 23]. In the present paper, we are concerned with this type of directed search where a user has to answer a specific question by accessing a web page and does this by entering a query and selecting a link from a result list to meet that information need. The key contribution to be made by adaptive, personalized systems in this area is to increase the relevance ordering of result lists, so that the most relevant sites are in the initial positions of the result list. This research goal has become more acutely important with the emergence of the mobile Internet and the shrinking screen real estate available for presenting information.

Over the past few years, several tools and techniques have been developed for evaluating the usability of websites [2, 5, 6, 7, 8, 18, 26, 27]. In general, this work takes an information foraging approach, seeing human information seeking as being analogous to animals foraging for food [25]. This approach casts users as followers of information scents when web searching [2, 5, 6, 7, 8, 26]. The basic idea is that users will assess the distal content - namely, the page at the other end of the link - using proximal cues, the snippets of text or graphics that surround a link [5, 6, 7, 8]. By comparing these cues with their information goal, the user chooses the link that best meets their current goal, namely, the link with the highest information scent [2, 5, 6, 7, 8]. Two main flavors of information foraging have been advanced for usability assessment, the Cognitive Walkthrough for the Web [2] and the InfoScent Bloodhound Simulator [8]. To date, these approaches have been mainly used to provide (semi)-automated usability assessments of websites. However, they also make predictions about link-choices made by users in first-click behavior.

The Cognitive Walkthrough for the Web (CWW) is a theory-based inspection method used to evaluate how well a website supports users navigation and information search tasks [2]. CWW uses Latent Semantic Analysis (LSA) [19, 20] to calculate the information scent of a link given a specific user information need. In essence, it works by calculating the LSA-derived similarity between the users information goal (i.e., the query) and the text surrounding a given link. CWW has been shown to successfully predict uninformative/confusing links on analyzed websites [2].

The InfoScent Bloodhound Simulator [8] is an automated analysis system that examines the information cues on a website and produces a usability report. Bloodhound uses a predictive modeling algorithm called Web User Flow by Information Scent (WUFIS) that relies on information retrieval techniques (i.e., TF.IDF analyses) and spreading activation to simulate user actions based on their information needs [6, 8]. In this paper, for reasons of space, we concentrate on CWW rather than Bloodhound. It should be pointed out that CWW does better than Bloodhound.

We were interested in applying these approaches to a developed adaptive system, the personalized search engine I-SPY [29, 30]. I-SPY is an example of an adaptive information retrieval system. See, Micarelli & Sciarrone [22] and Pierrakos et al, [24] for other related work on adaptive information filtering and Web personalization.

I-SPY implements an adaptive collaborative search technique that enables it to selectively re-rank search results according to the learned preferences of a community of

users. Effectively I-SPY actively promotes results that have been previously favored by community members during related searches so that the most relevant results are top of the result list [29]. I-SPY monitors user selections or *hits* for a query and builds a model of query-page relevance based on the probability that a given page will be selected by the user when returned as a result to a specific query [1, 13, 29, 30].

I-SPY has previously been shown to be capable of generating superior result rankings, based on its collaborative model of page relevance [1, 13]. For instance, we know from the results of a live-user trial, designed to compare the search effectiveness of I-SPY users against a control group, that I-SPY's promoted results are likely to be valuable to searchers. This study provided us with access to comprehensive search logs reflecting detailed search behavior information including the queries submitted by control and test groups, the results returned and promoted, the results selected, and their positions when selected. If the information scent approaches accurately capture user behavior we should find that the results chosen by people are indeed those with the highest information scent.

3 Evaluating I-SPY

In this section we describe key aspects of the I-SPY evaluation. Although we do not evaluate the I-SPY system in this paper, we have included relevant details regarding the I-SPY evaluation to illustrate the environment and conditions in which we are attempting to evaluate information scent predictions for first-click behavior. For further details regarding the I-SPY evaluation see, [1,14, 15, 29].

The I-SPY evaluation took place over two separate sessions and involved asking two separate groups of 45 and 47 Computer Science students, to answer a series of 25 questions on topics in Computer Science and Artificial Intelligence. In the first session, the I-SPY collaborative search function was disabled so the results presented to participants were drawn from a Meta search engine (using Google, AllTheWeb, Wisenut and HotBot). The students taking part in the first session served as a control group against which to judge the students taking part in the second session, where I-SPY's collaborative search function was enabled. When each person used I-SPY, they entered one or more query terms and were presented with a list of up to 20 results; consisting of a result number/rank, a title and a description/summary (or blurb).

A substantial amount of web search behavior data was generated from the I-SPY experiment. A total of 811 distinct queries were logged with 10,445 unique pages being returned in result lists. From these lists, a total of 427 unique pages were clicked on by users. All of this information was collected and archived for analysis in assessing the information foraging approaches to which we now turn.

4 Does CWW's Information Scent Predict First-Click Behavior

Cognitive Walkthrough for the Web [2] is a usability inspection technique for assessing information search tasks that uses Latent Semantic Analysis (LSA) [19, 20]. CWW calculates the information scent of a given piece of link text by finding its similarity (as computed by LSA) to the query terms used (or more commonly an elaborated description of the query). LSA captures patterns of co-occurrence between

words based on a text corpus analysis, in its most commonly used form, of general reading up to first-year college level in the US. LSA has been shown to predict some aspects of language comprehension and priming in various cognitive tasks [10, 11, 17, 19, 21]. In our analysis of the I-SPY data we looked at the information scent values computed from LSA for long/short queries compared against each and every piece of link text returned in the results lists (see Church, Keane & Smyth [9], for details).

4.1 Method

Data & Analysis Procedure. The analysis was performed on the data common to both I-SPY sessions. Limiting our dataset to just queries and urls common to both sessions of the I-SPY experiment provided us with valuable before and after ranking information as well as key web search behavior details. This common dataset consisted of 132 valid distinct queries and 2,571 results/urls.

Queries and results were sub-classified. For queries, we had short queries (the actual terms input by users) and long queries (an elaboration of the original question posed to the user). For results, we distinguished between (i) the title of the link alone, (ii) the blurb text alone (i.e., the summary text given back by the search engine) and (iii) both title and blurb. All six possible pairings for every unique query and result (2,571 distinct test items) were submitted to the LSA website [20]. In all, it took roughly 6 days of computing time to gather these scores. The short query - title case revealed the best results of all six possible pairings. One of the problems with LSA is that it does not contain many of the specialist terms used in Computer Science (we would assume that this is a general problem that would attach to any specialist domain). Hence, we filtered out those queries-result pairs that had large numbers of unknown terms to LSA (reduced set to 97 queries).

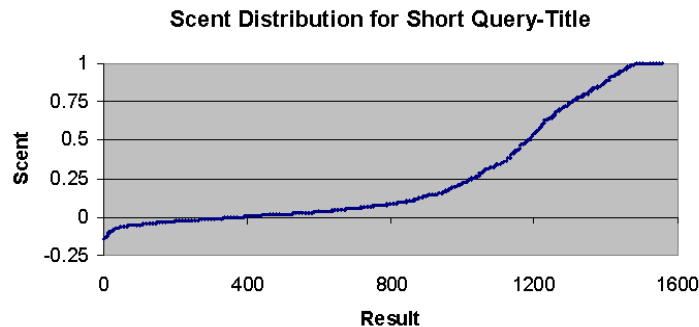


Fig. 1. Plot of the Scint Distribution for the Short Query-Title Pair in which the Results are Reordered by Scint

4.2 Results & Discussion

Overall, CWW does not do a good job of predicting the links clicked on by users in the I-SPY study. The correlations found are weak and, more damagingly, appear to mainly relay simple string-matching between the query and result.

Properties of the CWW Scent Values. Most of the scent values generated by LSA were low: the minimum score was -0.14 , the maximum was 1 ($M = 0.26$, $SD = 0.35$). Figure 1, above, shows the distribution of scent values for the short query-title pairing. In general, LSA generates high scent values when the text it receives has a high percentage of word-to-word matches and very few word-to-word mismatches (see our later analysis using string matching).

Does CWW's Information Scent Predict Link Choice ? The crucial question for CWW is whether its scent values predict the pages chosen by people in the study. To carry out this evaluation we extracted a subset of I-SPY data that included only the common hit data (i.e., links that were chosen by users, all of whom entered an identical query with the same question in mind). This set consisted of 110 distinct queries and 1,218 chosen links. If CWW predicts people's link choice then the hit score (i.e., the number of people choosing a given url) should correlate with its scent value. Unfortunately, this correlation is low. Table 1 shows the correlations between the short query and the three possible versions of the result (as title alone, blurb alone and title and blurb together). The correlations for long queries were worse, all ≤ 0.04 .

Table 1. Correlations between the Information Scent of the Short Query - Result Pair and the Hit Score

Result Type	Correlation	Classification
Short Query - Title	0.10	Very Weak
Short Query - Blurb	0.11	Very Weak
Short Query - Title + Blurb	0.13	Very Weak

Another perspective on this data can be gleaned from Figure 2 below, which shows scent values plotted by their hit scores for the best pairing (i.e., the short query – title + blurb). The most obvious conclusion from the graph is that many high-scented links have low hit scores and some low-scented links have high hit scores.

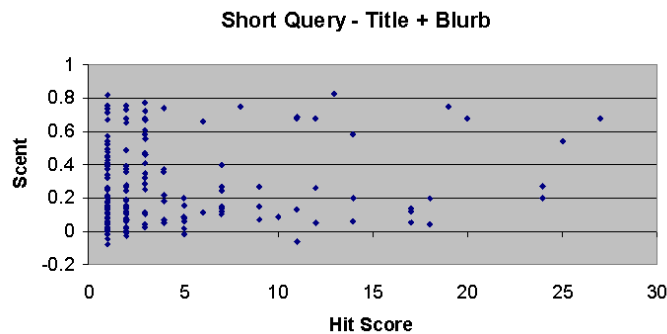


Fig. 2. Plot of the Scent Value by Hit Score for the Most Highly Correlated Query-Result Pair

One of the problems with this analysis is that some of the queries contain terms that are not in LSA's corpus¹. A fairer test would be to perform the same analysis for query-result pairs in which all the terms were known by LSA. However, even after removing all unsupported query-result pairs, the correlations did not improve appreciably, the highest being still low ($r = 0.15$).

Finally, to give CWW the best possible chance, we picked the most highly correlated cases (from the query-result matrix) to see what the best possible correlation could be. Figure 3 shows the result of this analysis. Note that each of the points here represent the scent value x hit score of a query-result pairing where that pairing could be any of the 6 possibilities (e.g., short query-title, long query-blurb, etc). In this best case, the correlation is better and moderate ($r = 0.5$). However, this good news must be tempered by the size and selective nature of the dataset.

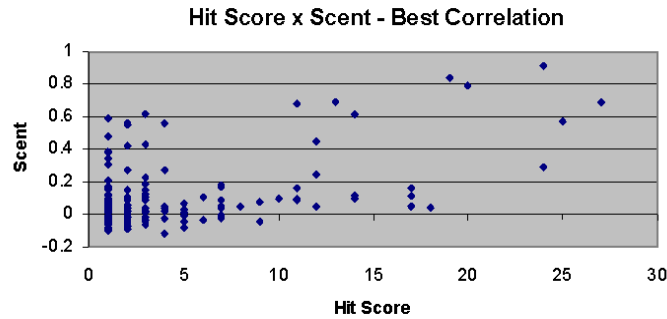


Fig. 3. Plot of the Scent Value by Hit Score for the Best-fit Correlations

Overall, we have to conclude that CWW can only generate reasonable predictions for a limited subset of queries (i.e., those for which it has a vocabulary) and even then it is not clear which one of 6 possible flavors of query-result pairing will best predict hits. This conclusion of limited appeal is further overshadowed by the possibility that CWW mainly appears to succeed by term matching.

Does CWW Succeed by Term Matching ? On the face of it, CWW's highest scent values seem to rely on string matching. To test this hypothesis, we applied a term matching model to the original dataset, using Tversky's Contrast Model of Similarity [32]. Tversky's model sees similarity as being based on the common and distinctive features of two items. The similarity between two objects a and b can thus be defined as,

$$S(a, b) = \theta f(A \cap B) - \beta f(A - B) - \alpha f(B - A) \quad (1)$$

where A represents the set of terms associated with a , B represents the set of terms associated with b , $A \cap B$ is the set of terms common to a and b , $A - B$ represents the set of features distinct in a , $B - A$ represents the set of features distinct in b . In this

¹ In these cases, LSA returns a value of N/A for the terms not in its corpus.

formula, f is usually a count of the features and θ , β and α are usually 1 or 0 (creating a family of models where one or another of the components can be cancelled out).

We then looked at the correlations between the scores produced by variants of the contrast model and CWW's scent values for the same items. We found that a simple term-matching model (i.e., the contrast model using only the common features component) was moderately-highly correlated with the CWW scent values ($r = 0.6$). Given that term matching is simpler and does not encounter the same vocabulary problems as CWW, it would appear to offer a better basis for predicting first-click behavior.

5 Bloodhound's Predictions for First-Click Behavior

We have carried out a similar analysis of the same data using the techniques developed in the InfoScent Bloodhound Simulator [8]. For reasons of space we cannot report these results here, suffice to say that the correlations are considerable worse than those found in CWW (see Church, Keane & Smyth [9], for details).

6 A Framework for Assessing First Clicks

Given our findings it is hard to escape the conclusion that, from a predictive perspective, we know very little about the basis of first-click behavior. As such, it would appear to be a good idea to step back from the problem and consider the main cognitive components of the context in which this task is being carried out. Card et al [5] have previously advanced a problem space of user's browsing behavior. However, we feel that their analysis is at a too fine-grained level to help us in this case and, maybe, really just provides us with a language with which to describe user behavior rather than a theory of the main parameters that impact that behavior. Therefore, in this section, we attempt to outline a general framework for understanding first-click behavior.

Broadly speaking, we can distinguish between the parts of the first click task that are represented in the user's head and those that are represented textually in the computer. The relevant data on the computer side are easily characterized and inspected. They include: the explicit question posed (if elaborated), the specific query terms used, the result lists returned, the ordering of those results, the distal pages to which these links refer and so on. On the human side, the relevant components are less easily characterized and not easily inspected. They include: the users' mental representation of the question, query and results, the user's background knowledge about the domain of the question, the user's knowledge of natural language, the user's knowledge of what ordered results entail, the user's similarity function for matching his/her information need to the presented result, the strategies the user normally employs when searching result lists, knowledge of previous searches and so on.

The key problem is that we do not have good techniques for acquiring and characterizing the knowledge that is brought to bear by users in choosing a link from a set of returned results. In theoretical terms, we need a well-developed cognitive model of this behavior. In practical terms, we need good proxies for this knowledge based on some analysis of the textual data we can explicitly enumerate in the task. In this sense a lot of the work to date can be characterized as proxies of varying goodness.

6.1 Some Proxies for User Knowledge

The usability methods employed here and many of the methods used to personalize and relevance-rank search engine outputs basically use some analysis technique that tries to approximate what people want using explicit data from the web context.

Link-Structure Analysis. Techniques that hinge on recommendation by analyzing link structure e.g., Google [4], essentially work on the assumption that the authority/relevance choices of a community of web-page builders, as indicated by their established links, will parallel the authority/relevance required by someone searching for a resource. The link structure created by the community is a proxy for the relevance ordering of the searching user.

Community-based Hits Analysis. Similarly, the techniques used by I-SPY, which hinge on analyzing the query-result choices of community users, also work on the assumption that what was good for others will be good for you. I-SPY's success is based on the closeness of this proxy to what the user is doing; using other people's choices to predict a new user's choice. In this respect, it is important to point out that I-SPY's builders assume that the community will be in some way representative of the user. This type of representative assumption is a familiar foundation for many approaches to lazy learning [31] and the degree to which it stands up in the context of I-SPY will depend largely on the focus of a particular community of searchers.

Corpus Analysis. CWW basically uses corpus analysis, based on LSA, to approximate a model of people's background knowledge for the words they use. On the face of it, this looks like a sound idea. But, other research has shown that LSA is not good at finding deep semantic similarity [12, 16]. This is exactly what our empirical analysis shows up. First, CWW fails because we fall off the edge of LSA's word knowledge (with specialist terms). Second, its generalization over word meanings is not powerful enough to be a good proxy to human knowledge.

Term-Frequency Analysis. Bloodhound makes heavy use of term frequency analysis in order to provide a proxy to people's knowledge of the domain. Our empirical studies show that varying the set of pages over which these values are computed (i.e., the domain) do not have a significant impact on the goodness of its predictions. It is hard to escape the conclusion that such term-frequency analyses are a poor proxy, on their own, for characterizing user knowledge.

7 Conclusions

Overall there are some positive and some negative conclusions to be made from the empirical analysis we have carried out here. Taking the bad news first, it is clear that current approaches using information scent do not do a good job of predicting the first clicks users make when presented with various results lists. In other words, we still need a good user model for this key behavior in web searching.

Happily, there are also a number of pieces of good news that we can take from this work. First, we outlined a methodology for the empirical evaluation of web search behavior. Second, we have shown that there are limitations to current information foraging theory that can be used productively to guide future theorizing. Third, with our presented framework, we have gained some perspective on the general nature of

first-click behavior. Fourth, we have seen that community-based hits analysis provides a reasonable proxy for first-click behavior, thus suggesting a fruitful direction for future work to characterize this behavior.

8 Acknowledgements

We thank the I-SPY research group at the University College Dublin, namely Jill Freyne, Evelyn Balfe, Peter Briggs and Maurice Coyle, who provided us with data from the I-SPY experiment. This work was funded by grants to the second and third authors from Science Foundation Ireland under Grant No.03/IN.3/I361.

9 References

1. Balfe, E., Smyth, B.: Case-Based Collaborative Web Search. In: Proceedings of the European Conference on Case-Based Reasoning, ECCBR'04, Springer (2004) Madrid, Spain.
2. Blackmon, M.H., Polson, P.G., Kitajima, M., Lewis, C.: Cognitive Walkthrough for the Web. In: Proceedings of the CHI 2002, ACM Press (2002) 463-470.
3. Broder, A.: A Taxonomy of Web Search. SIGIR Forum 36(2) (2002).
4. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings of the 7th International World Wide Web Conference, (1998) 107-117 Brisbane, Australia.
5. Card, S.K., Pirolli, P., Van Der Wege, M., Morrison, J.B., Reeder, R.W., Schraedley, P.K., Boshart, J.: Information Scent as a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability. In: Proceedings of CHI 2001, ACM Press (2001) 498-505 Seattle, WA.
6. Chi, E.H., Pirolli, P., Chen, K., Pitkow, J.: Using Information Scent to Model User Information Needs and Actions on the Web. In: Proceedings of CHI 2001, ACM Press (2001) 490-497 Seattle, WA.
7. Chi, E.H., Pirolli, P., Pitkow, J.: The Scent of a Site: A System for Analyzing and Predicting Information Scent, Usage, and Usability of a Web Site. In: Proceedings of CHI 2000, ACM Press (2000) 161-168 The Hague, The Netherlands.
8. Chi, E.H., Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., Robles, E., Dalal, B., Chen, J., Cousins, S.: The Bloodhound Project: Automating Discovery of Web Usability Issues using the InfoScent™ Simulator. In: Proceedings of CHI 2003, ACM Press (2003) Fort Lauderdale, FL.
9. Church, K., Keane, M.T., Smyth, B.: Evaluating Cognitive & User Models of "First-Click Behavior" in a Personalized Search Engine. User Modeling and User-Adapted Interaction, *Submitted*.
10. Connell, L., Keane, M.T.: PAM: A Cognitive Model of Plausibility. In: Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society, Erlbaum (2003) Hillsdale, NJ.
11. Connell, L., Keane, M.T.: What Plausibly Affects Plausibility. Concept Coherence and Distributional Word Coherence as Factors Influencing Plausibility Judgments. Memory & Cognition (2004).
12. French, R., Labiouse, C.: Four Problems with Extracting Human Semantics from Large Text Corpora. In: Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society, Erlbaum (2002) Hillsdale, NJ.
13. Freyne, J., Smyth, B.: Collaborative Search: A Live User Trial. In: Proceedings of the 26th European Conference on IR Research, ECIR'04, (2004) Sunderland, UK.

14. Freyne, J., Smyth, B.: An Experiment in Social Search. In: Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH-04, (2004) Eindhoven, The Netherlands.
15. Freyne, J., Smyth, B., Coyle, M., Balfé, E., Briggs, P.: Further Experiments on Collaborative Ranking in Community-Based Web Search. *AI Review: An International Science and Engineering Journal*, *In Press*.
16. Glenberg, A.M., Robertson, D.A.: Symbol Grounding and Meaning: A Comparison of High-Dimensional and Embodied Theories of Meaning. *Journal of Memory and Language*, 43 (2000) 379-401.
17. Kintsch, W.: Predication. *Cognitive Science*, 25 (2001) 173-202.
18. Kitajima, M., Blackmon, M.H., Polson, P.G.: A Comprehension-based Model of Web Navigation and Its Application to Web Usability Analysis. In *People and Computers XIV*, Springer (2000) 357-373.
19. Landauer, T.K., Dumais, S.T.: A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104 (1997) 211-240.
20. LSA Website: <http://lsa.colorado.edu>
21. Lund, K., Burgess, C., Atchley, R.A.: Semantic and Associative Priming in High-Dimensional Semantic Space. In: Proceedings of the 17th Annual Conference of the Cognitive Science Society, Erlbaum (1995) 660-665 Hillsdale, NJ.
22. Micarelli, A., Sciarrone, F.: Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. *User Modeling and User-Adapted Interaction*, 14(2-3) (2004) 159-200.
23. Morrison, J.B., Pirolli, P., Card, S.K.: A Taxonomic Analysis of What World Wide Web Activities Significantly Impact People's Decisions and Actions. In: Proceedings of CHI 2001, ACM Press (2001) 163-164 Seattle, WA.
24. Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos, C.D.: Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction*, 13(4) (2003) 311-372.
25. Pirolli, P., Card, S.K.: Information Foraging. *Psychological Review*, 106(4) (1999) 643-675.
26. Pirolli, P., Fu, W.-T.: SNIF-ACT: A Model of Information Foraging on the World Wide Web. In: Proceedings of the 9th International Conference on User Modeling, (2003) Johnstown, PA.
27. Polson, P.G., Lewis, C., Rieman, J., Wharton, C.: Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies*, 36 (1992) 741-773.
28. Salton, G.: *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley (1989).
29. Smyth, B., Balfé, E., Briggs, P., Coyle, M. and Freyne, J.: Collaborative Web Search. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03, Morgan Kaufmann (2003) 1417-1419 Acapulco, Mexico.
30. Smyth, B., Freyne, J., Coyle, M., Briggs, P., Balfé, E.: I-SPY: Anonymous, Community-Based Personalization by Collaborative Web Search. In: Proceedings of the 23rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer (2003) 367-380 Cambridge, UK.
31. Smyth, B., Keane, M.T.: Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. *Artificial Intelligence*, (102)2 (1998) 249-293.
32. Tversky, A.: Features of Similarity. *Psychological Review*, 84(4) (1997) 327-352.

Evaluating Intelligent Tutoring Systems with Learning Curves

Brent Martin and Antonija Mitrovic

Intelligent Computer Tutoring Group
Department of Computer Science, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
{brent,tanja}@cosc.canterbury.ac.nz

Abstract. The evaluation of Intelligent Tutoring Systems, like any adaptive system, can have its difficulties. In this paper we discuss the evaluation of an extension to an existing system that uses Constraint-Based Modelling (CBM). CBM is a student modelling technique that is rapidly maturing, and is suited to complex, open-ended domains. A problem with complex domain models is their large size, necessitating a comprehensive problem set in order to provide sufficient exercises for extended learning sessions. We have addressed this issue by developing an algorithm that automatically generates new problems directly from the domain knowledge base. However, evaluation of this approach was complicated by the need for a lengthy (and therefore uncontrolled) study as well as other unavoidable differences between the control and experimental systems. This paper presents the evaluation and discusses those issues, and the way in which we used learning curves as a tool for comparing disparate learning systems.

1 Introduction

Constraint-Based Modelling (CBM) [7] is an effective approach that simplifies the building of domain and student models in Intelligent Tutoring Systems (ITS). We have used CBM to develop SQL-Tutor [4], an ITS for teaching the SQL database language. SQL-Tutor supports learning in three ways: by presenting feedback when students submit their answers, by controlling problem difficulty, and by providing scaffolding information. Students have shown significant gains in learning after as little as two hours of exposure to this system [5].

SQL-Tutor contains a list of problems, from which one is selected that best fits the student's current knowledge state. In extended sessions with the tutor the system may run out of suitable problems. We have attempted to overcome this by developing a problem generator that uses the domain model to build new problems that fit the student model, and a problem selection algorithm that is tied more closely to the student model.

We evaluated the new algorithm against the existing SQL-Tutor system in a classroom setting, but had difficulty because there were other differences in the system besides what we were testing. Also, the evaluation was over an extended period in an uncontrolled environment, which made comparisons in student performance difficult. We were therefore unlikely to see any difference in outcome using purely subject pre- and post-testing.

In this paper we briefly introduce SQL-Tutor and the extensions to it. We then describe the experiment and its evaluation. We discuss the use of learning curves to analyse the performance of the two systems, and introduce a new measure, that of the initial learning rate. We compare this new measure to the usual ones of slope and intercept, and discuss why we think it is valid. Finally, we conclude why we think the initial learning rate may be a better measure for determining how much learning went on while students used each system.

2 SQL-Tutor

SQL-Tutor [4] teaches the SQL database query language to second and third year students at the University of Canterbury, using Constraint-Based Modelling (CBM). This approach models the domain as a set of state constraints, of the form:

If <relevance condition> is true for the student's solution,
THEN <satisfaction condition> must also be true

The relevance condition of each constraint is used to test whether the student's solution is in a pedagogically significant state. If so, the satisfaction condition is checked, which tests for valid constructs (syntactic constraints) or compares the solution to an ideal solution (semantic constraints). If a constraint succeeds, no action is taken; if it fails, the student has made a mistake and appropriate feedback is given. CBM has advantages over other approaches such as model tracing [1] in that the model need not be complete and correct in order to function. Further, it is well suited to domains where the number of alternative solutions is large or the domain is open-ended.

Ohlsson [7] does not impose any restrictions upon how constraints are encoded, and/or implemented. In SQL-Tutor we initially represented each constraint by a LISP fragment, supported by domain-specific LISP functions. In later versions we have used a pattern-matching algorithm designed for this purpose [3], for example:

```
(147
"You have used some names in the WHERE clause that are not from
this database."
(match SS WHERE (?* (^name ?n) ?*))
(or (test SS (^valid-table (?n ?t))
(test SS (^attribute-p (?n ?a ?t))))
"WHERE")
```

The above constraint checks for valid table and attribute names in the WHERE clause of the student's SQL statement. It is relevant if any names exist in the WHERE clause, and satisfied if each of these is either a valid table name or a valid attribute name.

The constraint language makes all of the logic for determining whether or not the constraint is satisfied transparent to the system, since it consists only of pattern matching and logical combination. Functions, such as “^valid-table” in the above example, are simply macros defined in the same language. We have used this property to develop a problem solver that can generate correct solutions from student

attempts by extracting (and unifying) the valid match patterns from each satisfied constraint and the *invalid* patterns from violated constraints. The algorithm then corrects the invalid fragments by unifying them against matched patterns of the *ideal* solution, and then combines the resulting solution fragments.

Unlike repair theory [8], we make no claim that this algorithm is modelling human behaviour. However, it has the advantage that a failed constraint means that the construct involved is *definitely wrong*: we do not need to try to infer where the error lies, so our algorithm does not suffer from computational explosion. For further details on this algorithm and the constraint language refer to [3].

3 Generating New Problems

In the original version of SQL-Tutor the next problem is chosen based on difficulty, plus the concept the student is currently having the most trouble with. The constraint set is first searched for the constraint that is being violated most often. Then the system identifies problems for which this constraint is relevant. Finally, it chooses the problem from this set that best matches the student's current proficiency level. However, there is no guarantee that an untried problem exists that matches the student model: there may be no problems for the target constraint, or the only problems available may be of unsuitable difficulty. Further, since the constraint set is large (over 650 constraints), many problems are needed merely to cover the domain. Ideally there should be many problems per constraint, in various combinations. In SQL-Tutor there is an average of three problems per constraint and only around half of the constraint set is covered. A consequence of this is that the number of new constraints being presented to the student tapers off as the system runs out of problems.

The obvious way to address this limitation is to add more problems. However, this is not an easy task. There are over 650 constraints, and it is difficult to invent problems that are guaranteed to cover all constraints in sufficient combinations that there are enough problems at a large spread of difficulty levels. To overcome this we have developed an algorithm that generates new problems from the constraint set. It uses the constraint-based problem solver described in Section 2 to create novel SQL statements, using an individual constraint (or, possibly, a set of compatible constraints) to provide a set of SQL fragments as a starting point. These are then "grown" into a full SQL statement by repeatedly splicing them together and unifying them against the syntactic constraint set until no constraints are violated. This new SQL statement forms the ideal solution for a new problem. The human author need only convert the ideal solution into a natural language problem statement, ready for presentation to the student.

We used the problem generation algorithm create a single problem per constraint, giving around 800 potential ideal solutions (note that the experimental version had an extra 250 constraints added). We then chose the best of these and converted them into natural language problem statements. On completion we had a new problem set of 200 problems, which took less than a day of human effort to build. Furthermore, when we plotted the number of new constraints applied per problem presented to a student, the point at which new problems failed to introduce any new concepts (i.e. previously unseen constraints) rose from 40 problems to 60, indicating that the new problem set increased the length of time that a student could fruitfully engage with the system.

The experimental system also used a different problem selection mechanism. In the control system, a new problem is selected by finding the constraint the student is currently violating most often and selecting the problem whose (static) difficulty rating is closest to the student's current proficiency level. In the experimental system we calculated the difficulty of each problem dynamically by computing its static difficulty from the number of constraints (and their complexity) relevant to it, and then added further difficulty for relevant constraints that the student is currently violating, and more still for constraints that the student has not yet encountered. Thus each problem is compared to the student's current knowledge. Once difficulties have been calculated for all problems, the one that is closest to the student's current proficiency is selected.

4 Evaluation

The motivation for Problem Generation was to reduce the effort involved in building tutoring systems by automating one of the more time-consuming functions: writing the problem set. Three criteria must be met to achieve this goal: the algorithm must work (i.e. it must generate new problems); it must require (substantially) less human involvement than traditional problem authoring; and the problems produced must be shown to facilitate learning to at least the same degree as human-authored problems. The first two were confirmed during the building of the evaluation system: the algorithm successfully generated problems, and the time taken to author the problem set was much less than would have been required for human authoring alone.

Additionally, if the method works, it should be possible to generate large problem sets, which will have the benefit of greater choice when trying to fit a problem to the user's current student model. We might therefore expect that given a suitable problem selection strategy, a system using the generated problem set would lead to faster learning than the current human-authored set, because we are better able to fit the problem to the student.

SQL-TUTOR was modified for this purpose and evaluated for a six-week period. The subjects were second year University students studying databases. The students were broken into three groups. The first used the current version of SQL-TUTOR, i.e. with human-authored problems. The second group used a version with problems generated using the algorithm described. The third group used a version containing other research (student model visualisation) that was not relevant to this study. Before using the system, each student sat a pre-test to determine their existing knowledge and skill in writing SQL queries. They were then free to use the system as little or as often as they liked over a six week period. Each student was randomly assigned a "mode" that determined which version of the system they would use. At the conclusion of the evaluation they sat a post-test.

When the study commenced, 88 students had signed up and sat the pre-test, giving sample sizes of around 30 per group. During the evaluation this further increased as new students requested access to the system. At the conclusion of the study some students who signed up had not used the system to any significant degree. The final groups used for analysis numbered 24 (control) and 26 (experimental) students each. The length of time each student used the system varied greatly from not using it at all

to working for several hours, with an average of 2½ hours. Consequently, the number of problems solved also varied widely from zero to 98, with an average of 25.

There are several ways we can measure the performance of the system. First, we can measure the means of the pre-test and post-test to determine whether or not the systems had differing effects on test performance. Note, however, that with such an open evaluation as this it is dangerous to assume that differences are due to the system, since use of the system may represent only a portion of the effort the student spent learning SQL. Nevertheless, it is important to analyse the *pre-test* scores to determine whether the study groups are comparable samples of the population. This was found to be the case. There was similarly no significant difference in post-test scores, as we might expect.

Second, we can plot the reduction in error rates as the student practices on each constraint, or the “learning curve”. Each student’s performance when measured this way should lead to an exponential curve or so-called “Power law” [2, 6], which is typical when each underlying object being measured (in this case a constraint) represents a concept being learned. The steepness of this curve is an indication of the speed with which the student, on average, is learning new constraints. Since each constraint represents a specific concept in the domain, this is an indication of how quickly the student is learning the domain. We can then compare this learning rate between the two groups.

Finally, we can look at how difficult the students found the problems. This is necessary to ensure that the newly generated problems did not negatively impact problem difficulty (either by being too easy or too hard). There are several ways we can do this. First, we can measure how many attempts the student took on average to solve a problem and compare the means for the control and test groups. Second, students were permitted to abort the current problem and were asked to cite one of three reasons: it was too easy, it was too hard or they wanted to try a problem of a different type. If the proportion of problems aborted rises, or the ratio of “too hard” to “too easy” problems is further from 1:1 than the control group, we might conclude that problem difficulty has been adversely affected.

In this study, we measured all of the above. We used the software package SPSS to compare means and estimate power and effect size, and Microsoft Excel to fit power curves. We measured problem difficulty both subjectively and objectively: we obtained subjective results by logging when students aborted a problem and recording their reason. Any significant difference in the ratio of “too easy” to “too hard” responses would suggest we have adversely affected problem difficulty. Further, the percentage of problems aborted should not rise significantly. Next, we measured the number of attempts taken to solve each problem, and the time taken on each attempt. There was no significant difference in any of these comparisons. In other words, the students appeared to have found the level of difficulty of each problem about the same for both systems.

4.1 Learning Curves

Since the general tests failed to show any difference between the two groups, we turned to learning curves as a means of evaluating more closely what was occurring *while the system was being used*. We observed the learning rate for each group by plotting the proportion of constraints that are violated for the n th problem for which

this constraint is relevant. This value is obtained by determining for each constraint whether it is correctly or incorrectly applied to the n th problem for which it is relevant. A constraint is correctly applied if it is relevant at any time during solving of this problem, and is *always* satisfied for the duration of this problem. Constraints that are relevant but are violated one or more times during solving of this problem are labelled erroneous. The value plotted is the proportion of all constraints relevant to the n th problem that are erroneous.

If the unit being measured (constraints in this case) is a valid abstraction of what is being learned, we expect to see a “power curve”. We fitted a power curve to each plot, giving an equation for the curve. Note that as the curve progresses learning behaviour becomes swamped by random erroneous behaviour such as slips, so the plot stops trending along the power curve and levels out at the level of random mistakes. This is exacerbated by the fact that the number of constraints being considered reduces as n increases, because many constraints are only relevant to a small number of problems. We therefore use only the initial part of the curve to calculate the learning rate. Fig. 1 shows such plots, where each line is the learning curve for the entire group on average, i.e. the proportion of constraints that are relevant to the first problem that are incorrectly applied *by any student in the group*. The cut-off was chosen at $n=5$, which is the point at which the power curve fit for both groups is maximal. Note that learning curves are also exhibited when the data is plotted for a single student, although there is large variance between faster and slower learners, and the quality of the curves is often lower due to the small amount of data available. Similarly, plotting learning curves on a per-constraint basis (averaged over all students) gives power laws where the slope indicates the ease with which this constraint is learned, which is an indication of the quality of the constraint. For example, constraints that span more than one concept will produce a poor curve.

Both plots exhibit a very good fit to a power curve. The equations give us the Y intercept and slope for a log-log graph of constraint performance. In this case the experimental group had a Y intercept that was around *twice* that for the control group,

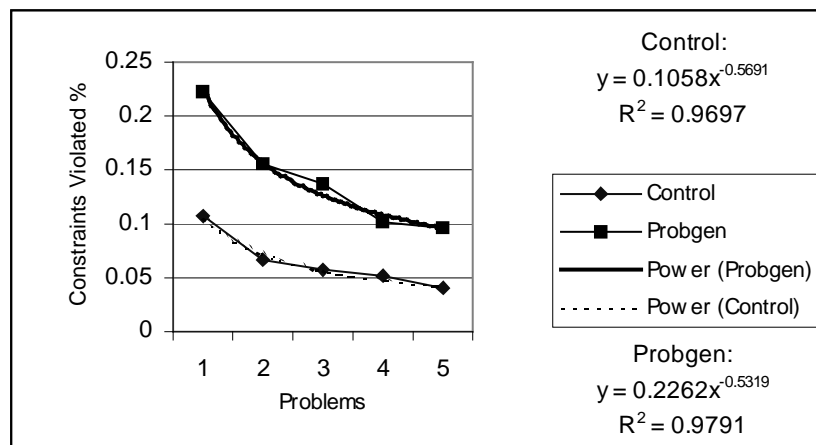


Fig. 1. Learning Performance

but a slightly lower slope (0.53 versus 0.57). Fig. 2 shows log-log plots for the same data.

4.2 Learning Curve Slope

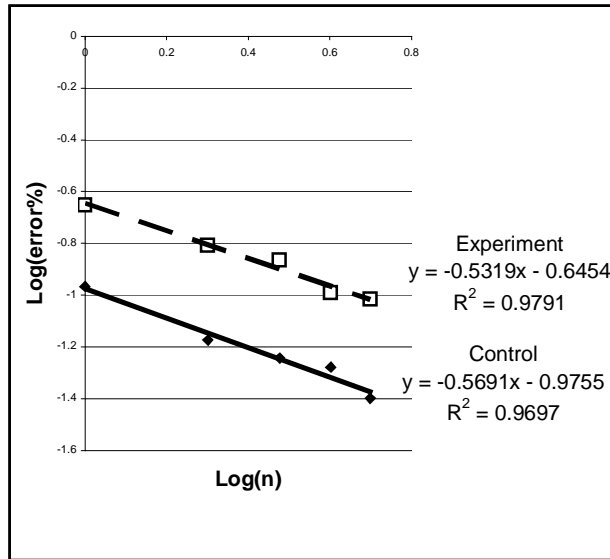


Fig. 2. Log-Log plots for the two groups.

The slope of a learning curve represents the *learning rate*, i.e. the rate at which the student decreased their errors with practice. There are several reasons why this might differ between the two groups: the students may have different average learning abilities; the constraints may represent domain concepts to a greater or lesser degree; the concepts may be introduced at more or less opportune times (and thus the students may be more or less receptive to learning the constraints at

that time).

The first difference (learner ability) was eliminated because the average pre-test scores were not significantly different for the two groups. Differences in the constraint sets were a possibility because the experimental system was a rewritten version of the control system using the new constraint representation: although the constraint set for the experimental group was based on that for the control group, a large number of modifications had been made, including the rewriting of many constraints that were always relevant, such that they were now only relevant in appropriate situations. New constraints were also added.

We tested the effect of these changes by recalculating the curves in two ways. First, we removed the data for all constraints that were trivially true (which occurred only in the control system's constraint set) and replotted the learning curve for the control group. Second, we took the raw student solutions from logs for the control group and evaluated them using the new constraint set. This latter method is a trade-off: it gives us the student's performance based on evaluation by the new constraints, but where the student still received feedback from the old constraint set.

In both cases the slope varied according to which method we used. Table 1 lists the results. The learning curve slope thus appears to be very sensitive to how the student's performance is being measured, and hence is not a suitable measure for comparing disparate systems.

Table 1. Learning curve slope for different constraint sets

Measuring method (Control Group)	Control Slope	Experiment Slope
Original constraint set	0.57	0.53
Exclude trivially true	0.31	0.53
Experimental constraint set	0.44	0.53

4.3 Y Intercept and Initial Learning Rate

The other measure of a learning curve is the Y intercept, which gives us the initial error rate. This alone is not a useful measure because it only indicates the student's initial performance, but does not show any effects of learning. However, the *slope* at $X=1$ (equal to the Y intercept multiplied by the log-log slope) does take learning into account: it indicates the raw improvement in performance achieved by a student between when they are first exposed to a constraint and when they have received feedback on it once. We therefore hypothesised that this is a better measurement of performance between disparate systems because it takes into account both the student's learning rate and the amount of unlearned material they are being exposed to as a percentage of the original constraint set. We expected that this calculation would be relatively invariant to changes in the constraint set, because it normalises out differences in the way the student's learning performance is measured.

Table 2 lists the results for the three ways that the students' performance was measured, i.e. using the original constraint, the same constraint set with trivial constraints excluded, and the new (experimental) constraint set. They show that, although the initial learning rate does differ depending on the constraint set used, the difference is much smaller than when the log-log slope is considered.

These results show that the slope at $X=1$, or initial learning rate, is almost twice as high for the experimental group. We checked for statistical significance by plotting learning curves for each individual student and comparing the means of the initial learning rate for the two groups. The difference was statistically significant at $\alpha=0.05$ ($p=0.01$). This suggests that the raw amount learned by the experimental group was higher than for the control group, which we attributed to either, or both, the increased number and range of problems available and an improved problem selection mechanism. Both of these could lead to a larger number of unknown constraints being presented to the student that were within their zone of proximal development [9]. Thus whereas the new problem set and selection method might not increase the student's learning rate, nevertheless it engages them in a larger volume of learning and therefore reduces the time taken to master the material.

Table 2. Initial slope for different constraint sets

Measuring method (Control Group)	Control Init. Slope	Exp. Init. Slope
Original constraint set	0.06	0.12
Exclude trivially true	0.05	0.12
Experimental constraint set	0.08	0.12

5 Discussion

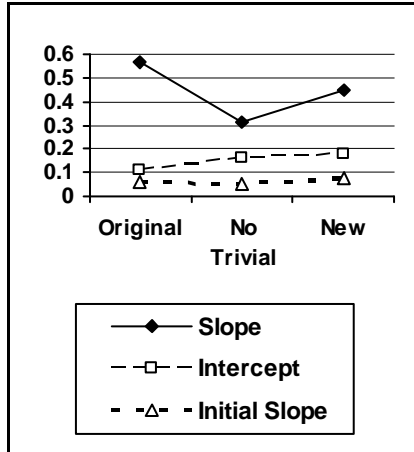


Fig. 3. Learning curve statistics.

The two main issues with this study were how to measure learning differences between two dissimilar systems, and which measure to use to demonstrate differences that we were actually interested in. Learning rate (based on learning curve slope) was considered but discounted because it measures how fast the student is learning, but is relative to how their performance is being measured. Thus, it can be expected to vary between different systems. Further, it does not give any indication of the size of the learning task: the same numbers may be obtained for a student who knows 95% of the material they are initially presented with as one who knows

only 10%, and is thus learning a greater volume of material at once. In both cases the ease with which the student learns the material is the same, but the amount they are learning varies widely. On the other hand, the Y intercept gives us the amount of learning being undertaken (i.e. what percentage of the presented material the student is trying to learn) but not the rate. However, the *initial* learning rate (slope at X=1) combines both the size of the learning task and the student's learning performance.

This new measure appears to be relatively invariant to the way student performance is measured, the issue in this case being differences in the constraint set. Fig. 3 plots how the measures of slope, Y intercept and initial slope vary with the constraint set used. Of the three, the initial slope varies the least. This is intuitively expected, and is illustrated in Fig. 4. These three curves are raw data for the control group, the same data multiplied by two, and again with a constant (0.5) added. The first two curves

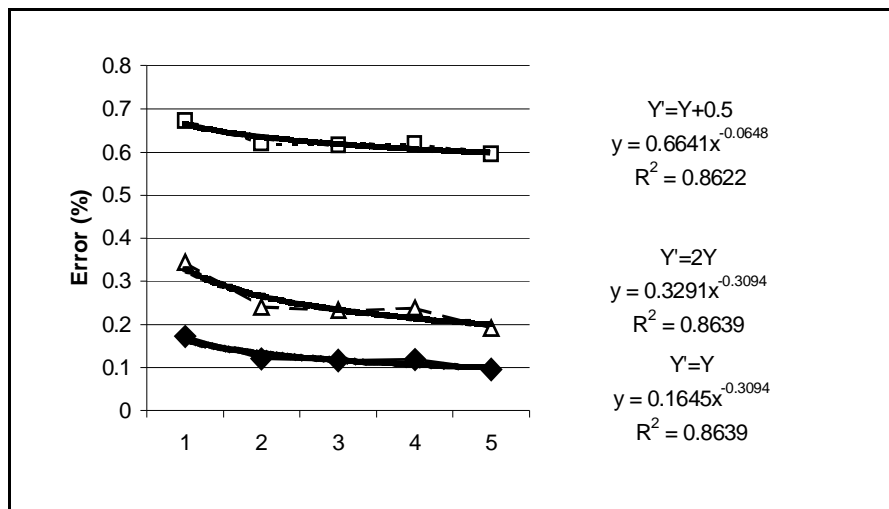


Fig. 4. Variations in power curve slope from artificial manipulation

have the same slope, even though the proportion of constraints being learned is twice as high for the second curve. However, the *initial* slope for the two curves has a ratio of 2:1, representing this effect. In contrast, the third curve is the same data with a constant amount added, which might represent a constraint set that includes trivially satisfied constraints. In this case the slope of the modified curve is dramatically lower (0.06 versus 0.3), even though the student's behaviour is unchanged. In contrast the initial learning rate is only slightly lower (0.43 versus 0.5).

6 Conclusions

This paper identified the problem of measuring student performance with intelligent tutoring systems when the systems being compared do not measure student performance in the same way. We showed that learning curves might still be used to compare such systems, but that the traditional measure of slope is not suitable because it varies with the method used to measure performance. We suggested a new statistic, that of the initial learning rate, which is produced by calculating the slope at $X=1$. We argued that this measure encompasses both the learning rate and the size of the learning task, and hence tells us more about the difference in performance between the two systems, because modifications to some aspects of a learning system (such as the problem set and the problem selection method) may alter the amount of material a student is learning at any time. We also showed that initial learning slope appears to be relatively robust against differences in the method used to measure the student's performance because it is normalised with respect to the student's initial performance.

Acknowledgement

This research was supported by the University of Canterbury grant U6430.

References

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., and Pelletier, R., *Cognitive Tutors: Lessons Learned*. Journal of the Learning Sciences, 1995. **4**(2): pp. 167-207.
2. Anderson, J.R. and Lebiere, C., *The atomic components of thought*. 1998, MahWah, NJ: Lawrence Erlbaum Associates.
3. Martin, B. and Mitrovic, A. *Tailoring Feedback by Correcting Student Answers*. in *Fifth Int. Conf. on Intelligent Tutoring Systems*. 2000. Montreal: Springer. pp. 383-392.
4. Mitrovic, A. *Experiences in Implementing Constraint-Based Modeling in SQL-Tutor*. in *4th Int. Conf. on Intelligent Tutoring Systems*. 1998. San Antonio, Texas: Springer. pp. 414-423.
5. Mitrovic, A. and Ohlsson, S., *Evaluation of a Constraint-Based Tutor for a Database Language*. Int. J. Artificial Intelligence in Education, 1999. **10**: pp. 238-256.
6. Newell, A. and Rosenbloom, P.S., *Mechanisms of skill acquisition and the law of practice*, in *Cognitive skills and their acquisition*, J.R. Anderson (ed.) 1981, Lawrence Erlbaum Associates: Hillsdale, NJ. pp. 1-56.
7. Ohlsson, S. *Constraint-Based Student Modeling*. in *NATO Advanced Research Workshop on Student Modelling*. 1991. Ste. Adele, Quebec, Canada: Springer-Verlag. pp. 167-189.
8. VanLehn, K., *On the Representation of Procedures in Repair Theory*, in *The Development of Mathematical Thinking*, H.P. Ginsburg (ed). 1983, Academic Press: New York. pp. 201-252.
9. Vygotsky, L.S., *Mind in society: The development of higher psychological processes*. 1978, Cambridge, MA: Harvard University Press.

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
CACM				
Initial run	0.065	0.091	0.067	0.095
Feedback run	0.12	0.2	0.090	0.223
MEDLINE				
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

	CACM		Medline	
Experiments	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)

An Example of Evaluation applied to a course adapted to learning styles of CHAEA's test

M^a del Puerto Paule Ruiz, Juan Ramón Pérez Pérez, Martín González Rodríguez

HCI Group Research
Dpto. Computer Science of University of Oviedo
C/Calvo Sotelo s/n. 33007 Oviedo
{paule, jrpp, martin}@uniovi.es

Abstract: This paper shows the results of an evaluation of a course adapted to learning styles of CHAEA's test. It is a comparative analysis between an adapted course and a course without adaptation also.

1 Introduction

Normally, the courses published in the Web are thought to get the learning of the students that visit the web site, but the majority of these courses do not include adaptation, which means that the student has to adapt himself to the Course and not the Course to the user. When the user has to adapt to the Course, often the result is not as desired, since the user is not comfortable and will probably not visit the Web site again and the initial goals of learning and diffusion are not carried out.

The adaptation is necessary, but the question is: What type of adaptation: Lexical [1], syntactic [4] or conceptual? The conceptual level seems the most appropriate [7], because this type includes cognitive parameters. These cognitive parameters are very important in the design of the on-line courses.

2 CHAEA's Test

There are a lot of cognitive parameters, but, one of the most important is the learning style. The learning styles are the way of thinking, the way of processing the information, and the way of learning each individual student has. There are a lot of classifications of the learning styles, but the classification selected in this paper is the CHAEA's test.

This test offers an acceptable reliability and validity that has been proved in Spanish Universities [2] and returns the preferences of the student at the time of learning. There are four styles in this classification: Theoretician, Activist, Reflexive and Pragmatic. The test returns a value between 0 and 20 for each style. With these values we get know the learning style. For example, if the student gets 20 in Activist style, he/she is Activist and she/he is going to learn like an Active learner.

Each style has its own characteristics and particularities that are clearly defined by the author [2] but we are going to explain some characteristics that we consider important in order to understand the adaptation [3].

Firstly, the adapted course has been implemented for the Theoretician and Activist learning styles. The selection of these learning styles is because the Theoretician and Activist are the base of Pragmatic and Reflexive learning styles.

Secondly, the Theoretician student likes the theory and she/he doesn't learn with examples or exercises. She/he learns in an inductive way and the contents have to be organized by concepts; however the Activist students like very little the theory and they prefer the exercises. They need a continuous feed-back.

3 Evaluation

In this experiment a Course about HTML is designed [5]. This course is very basic and it consists of six lessons: First Page with HTML, Headings, Paragraphs, and Design with Style, List and Links. It has been adapted for the Theoretician and Activist Users.

3.1 Formulation of hypothesis:

There are two questions:

1. Is the learning with adaptation better than the learning without adaptation?
2. Does the way of evaluating depend on the learning style?, if, do theoretician students resolve only theory questions? Do activist students resolve only active questions?

The experiment has to prove the following hypothesis:

1. The learning with the Adaptation Model is more effective than the learning without the Adaptation Model.
2. The evaluation of the knowledge does not depend on the learning style. The most important thing is the learning itself and not how this learning is evaluated.

3.2 Identification of variables

Two variables have been selected to develop the experiment: The learning style (independent variable), and the result of the evaluation of knowledge (dependent variable).

3.3 Population and Sample

The population consists of students of the subject Information Systems of the Business School of University in Oviedo.

The sample is composed of 54 students. 27 of them are theoreticians and 27 of them are activists. For all of them it is the first time that they have been registered in the subject and they do not know anything about HTML.

There are three groups: A Control group, an experimental group and the Non_Adapted_Test Group. These groups are homogeneous in their composition, so, there are nine theoretician and nine activist students in each group. The distribution of students is random in each group.¹

3.4 The design of th experiment

There are two types of courses:

1. An adapted course: This course has different interfaces depending on the student [6]:
 - The Theoreticians have more theory than exercises and the theory is organized in concepts.
 - The Activists have more exercises and more examples than theory.
2. A course without adaptation: This course offers the same interface with theory, examples and exercises for each lesson.

There are also two types of tests:

1. Test with questions adapted to learning styles
2. Test with questions without specific adaptation, so, there are questions of both learning styles.

The experimental sessions are developed in the Business School of University of Oviedo. Each session is 110 minutes long.

These sessions have four parts:

1. Attitude Test².
2. Chaea's Test.
3. Surfing on the Course
4. Acquired knowledge test.

The 1st and 2st parts are equal for all groups, but the 3st and 4st parts are different depending on the group.

The control group students have to surf the course without adaptation and they have to do a test adapted to the learning style.

The experimental group students have to surf the adapted course and they have to do a test adapted to the learning style.

The Non_Adapted_test group have to surf the adapted course and they have to do a test without adaptation.

The students of these groups are different.

The test is formed by 15 multiple answer questions. At the end of it, the students know the score.

The test has 3 questions for each lesson of the course.

¹ Each group has majority of girls than boys because more girls than boys registered in the subject Information Systems.

² This test is in bases om Likert Scale with numeric answers where 1 represents the lowest agreement and 5 represents the highest agreement.

Each correct answer scores 1 point and each incorrect answer scores -1 point and the non answered question scores 0 points.

At the beginning of the session the student is told that it is part of a teaching quality evaluation of the University of Oviedo, so the student is not conditioned by the test.

3.5 Results

With this experiment, information is about the attitude and what knowledge the students acquire after surfing the course.

With this information, it is possible to determine if the adaptation has an influence on the learning and if the way of evaluating depends on the learning style.

The statistical software SPSS is used to get this analysis.

Firstly, it is necessary to determine if the groups had a similar one on the attitude: A Tstudent is applied to prove this³.

Secondly, it is necessary to check if the results of the control group are better than the results of the experimental group.

Thirdly, we must compare the results of the experimental group (adapted test) and the Non_Adapted_test group (test with a mixture of theoretician and activist questions).

The results of this analysis are shown in the appendix.

In the first part of this analysis, all groups (Control Group, Experimental Group and Non_Adapted_Test group) have a normal distribution for each learning style. Besides, the attitude has a normal distribution in Control Group and Experimental Group.

In the second part of this analysis, the control Group and the Experimental Group have a homogeneous variance and, the Experimental Group and the Non_Adapted_Group have a homogeneous variance too. The Attitude variance is homogeneous in the Control Group and Experimental Group. So, it is possible to apply the TStudent to get the differences and the improvements.

The application of TStudent in the Control Group and Experimental Group returns an improvement, so, the scores obtained in the Experimental Group are slightly better than the scores obtained in the Control Group. This result proves the first hypothesis: There is an increment of learning with the adaptation.

The application of Tstudent in the Experimental Group and Non_Adapted_test returns that the difference is not significant, so, the evaluation is not important, the most important thing is the learning. When a student learns, she/he can answer any type of questions. The evaluation is independent of learning style.

³ TStudent compares independent samples (for example, the different groups scores) and it is used to know if the differences are significant. The TStudent's proof requires normal distributions to apply it. The Shapiro-Wilk test is used to get this, it is also necessary that the variances are homogenous; the Levene's Proof is used to do this verification.

4 Conclusions

The evaluation offers a determinant conclusion: adaptation of contents versus adaptation in the evaluation of contents. The most important thing is the adaptation of contents, and not the adaptation in evaluation of the same ones.

Learning increases with the adaptation of contents, but there are not differences between the evaluations of adapted or not adapted contents. The process of teaching-learning has to focus on the adaptation of contents and not on the evaluation of the contents.

A course that respects the learning style makes the learning more efficient. Internet is the most adequate way to do this adaptation. The contents can be shown in a personal way to every user of the net: this is the power of the Web, since the same contents it can come in different ways to different individuals and this is a fundamental idea : The same knowledges explained in a different way depending on the characteristics and particularities of every user.

Acknowledgements. This research is supported by University of Oviedo, Proyecto MB-04-534-4.

Appendix

Note: Significance level of 95% (0, 05)

1. Normal Distribution for Control Group in Final Score (Shapiro-Wilk)

Learning Style	W	Sig
Activists	0,946	0,637>>0,05 → Distribution can be normal
Theoreticians	0,871	0,163>>0,05 → Distribution can be normal

2. Normal Distribution for Experimental Group in Final Score (Shapiro-Wilk)

Learning Style	W	Sig
Activists	0,854	0,092>>0,05 → Distribution can be normal
Theoreticians	0,854	0,091>>0,05 → Distribution can be normal

3. Normal Distribution for Not Adapted Group in Final Score (Shapiro-Wilk)

Learning Style	W	Sig
Activists	0,828	0,064>>0,05 → Distribution can be normal
Theoreticians	0,828	0,064>>0,05 → Distribution can be normal

4. Normal Distribution of Attitude in the Control Group in Final Score (Shapiro-Wilk)

Learning Style	W	Sig
Attitude	0,924	0,223>>0,05 → Distribution can be normal

5. Normal Distribution of Attitude in the Experimental Group in Final Score (Shapiro-Wilk)

Learning Style	W	Sig
Attitude	0,924	0,223>>0,05 → Distribution can be normal

6. Homogeneity of variance between the scores of Control Group and Experimental Group for each learning style

Learning Style	W	Sig
Activists	3,115	0,097>>0,05 → Variances are homogenous
Theoreticians	4,283	0,006>>0,05 → Variances are homogenous

7. Homogeneity of variance between the scores of Experimental Group and Not Adapted Test Group for each learning style

Learning Style	W	Sig
Activists	0,142	0,711>>0,05 → Variances are homogenous
Theoreticians	0,139	0,715>>0,05 → Variances are homogenous

8. Homogeneity of variance between the scores in the Likert Scale for the attitude

Levene	Sig
0,002	0,899>>0,05 → Variances are homogenous

9. TStudent for Control Group and Experimental Group

Learning Style	gl	Sig
Activists	16	0,001<<0,05 → The significant improvement
Theoreticians	16	0,002<<0,05 → The significant improvement

10. TStudent for Experimental Group and Not Adapted Test Group

Learning Style	gl	Sig
Activists	16	0,567>0,05 → The different is not significant
Theoreticians	15	0,567>0,05 → The different is not significant

References

1. Abascal J., Arrue M., Fajardo I., Garay N., Tomás J. Use of Guidelines to automatically verify web accessibility. Universal Access in the Information Society.

Special Issue on "[Guidelines, standards, methods and processes for software accessibility](#)" (Guest editors: Gulliksen J., Harker S., Vanderheiden G.). Accepted for publication.

2. Alonso, C.M., Gallego, D.J., Honey, P. "Los Estilos de Aprendizaje". 4ª Edición. Ediciones Mensajero.
3. Del Moral Pérez and et al. "Rur@Inet: Un espacio para la teleformación y el trabajo colaborativo". In 2nd European Congress on Information Technologies in Education and Citizenship: a critical insight, 2002.
4. Gilbert, J.E., Han, C. Y. "Adpating instruction in search of 'a significant difference'". Journal of Network and Computer Applications, 22.
5. Gross Begoña. "Diseños y programas educativos". ISBN: 84-344-2604-8, 2002
6. Paule Ruiz, Mª Del Puerto, Ocio Barriales, Sergio, Pérez Pérez, Juan Ramón, González Rodríguez, Martín. Feijoo.net. "An Approach to Personalized E-learning Using Learning Styles". Web Engineering, International Conference, ICWE03. Lecture Notes in Computer Science 2722. ISBN: 3-540-40522-4 Springer Verlag Berlin . ISSN: 0302-9743, 2003
7. González Rodríguez, Martín, López Pérez, Benjamin, Páule Ruíz, María Del Puerto, Pérez Pérez, Juan Ramón. "Dynamic generation of Interactive Dialogs Based on Intelligent Agents". Lecture Notes in Computing Science Vol. 2347, Springer, May 2002. Pp. 564 - 567. ISSN 0302-9743, 2002

Evaluating a General-Purpose Adaptive Hypertext System

Chris Staff

University of Malta,
Department of Computer Science and AI,
Msida MSD 06, Malta

Abstract. We describe the evaluation process of HyperContext, a framework for general-purpose adaptive and adaptable hypertext. In particular, we are interested in users' short-term, transient, interests. We cannot make any prior assumptions about a user's interest or goal, as we do not have any prior knowledge of the user. We conducted evaluations on two aspects of HyperContext. One evaluation was completely automated, and the other involved participants. However, the availability of a test collection with value judgements would be a considerable asset for the independent and automated evaluation of adaptive hypertext systems in terms of cost, reliability of results, and repeatability of experiments.

1 Introduction

HyperContext is a framework for adaptive and adaptable hypertext [8], [9]. We are currently using the HyperContext framework as part of the University of Malta's contribution to the Reasoning on the Web with Rules and Semantics (REWERSE) FP6 Network of Excellence¹.

HyperContext focuses on building and maintaining a short-term user model to provide adaptive navigation support. We begin a user session with an empty user model and we add to the model as a user navigates through hyperspace and interacts with the system.

A proof of concept HyperContext application has been evaluated. We had devised an evaluation strategy for HyperContext in 1999. However, due to a number of reasons, including hardware failure, the original evaluation strategy was abandoned. We eventually settled on a partially automated approach that did involve some participants, but which was less reliant on human participants.

We are satisfied that the results of the automated evaluation show that the adaptive features of HyperContext can guide users to relevant information. We feel that our automated evaluation benefited from the fact that HyperContext assumes an initially empty user model that is then populated during short interactions with the system. Part of the evaluation involved showing users a series of documents (representing a short path through hyperspace) followed by two other documents in a random sequence. One of the two documents was recommended

¹ staff.um.edu.mt/mmon1/research/REWERSE/

by HyperContext using a user model that would have been generated had the user actually followed the path through the first 5 documents in a HyperContext hyperspace. The other document was also a recommended document, but the user model used to make the recommendation was derived in a different way. We are able to demonstrate that the second recommendation is based on a user model built on a Web-based, rather than a HyperContext-based, hyperspace. The evaluation is similar to an approach using with- and without-adaptive functionality [6], but we show that the without-adaptive functionality system is equivalent to the World Wide Web. The results of the evaluation are reported extensively in [8] and [9]. In this paper we concentrate on reporting the evaluation *process* and our opinion on its suitability for the evaluation of adaptive hypertext systems.

2 Objectives of the Evaluation

Before we discuss HyperContext and the evaluation strategies, we present our motivation and objectives for evaluating HyperContext. An adaptive hypertext system may use adaptive navigation techniques to guide users to relevant information in hyperspace [2]. As HyperContext utilises adaptive navigation techniques almost exclusively (there is limited support for adaptive presentation, but this was not the focus of our research), we expected that a HyperContext user would find relevant information faster than a user using a non-adaptive equivalent, as HyperContext would recommend links and paths to users, assuming that the user model accurately reflected the user's needs and requirements.

HyperContext is a general-purpose system for use in a heterogeneous information space, such as the WWW. Consequently, unlike an educational hypertext system, we cannot make certain assumptions about our users. For instance, the goal of a user of an educational system is likely to be to increase his or her knowledge of the subject contained within the system. As the domain is restricted, it is possible to pre-test or "interview" the user to initialise the user model with useful information. The users of a general-purpose hypertext system that focuses on collection of short-term information are not so helpful. A short-term user model is likely to be at its most useful when the user is navigating through territory with which he or she is unfamiliar and when the user's interest in the information is significant but transient. For instance, a user may have some task to perform and some information is required to perform that task. Although the completion of the task is dependent on obtaining the information, the user's interest in it lasts only as long as it takes to complete that aspect of the task. What motivates us is the challenge of recommending useful links (i.e., links that are likely to lead the user to relevant information) when we initially know little or nothing about the user's interests, goals, and expertise. However, motivating evaluation participants to the degree that they will search for information that they know little about but really need under evaluation conditions is hard. Either the prototype software under evaluation will need to be robust enough to use on the Web at large (in which case participants can use the system in their

own time), or a smaller Web space will be converted for use with the hypertext system (so that HTML pages, for instance, will be free from error), in which case the chances of finding adequately motivated participants is greatly reduced.

For our evaluation, we converted part of the World Wide Web Consortium's (W3C) website² to a HyperContext hyperspace. We chose the W3C site because it is about Web standards ranging from HTML to Web-HCI issues, so we reasoned that the site was designed to be easy to use, consistent, and relatively free from (HTML) errors, which would ease processing. An explanation of what is involved in the conversion is given in section 3. We also show in subsection 4.1 that without the adaptive features provided by HyperContext, the converted site is equivalent to the original Web site.

3 Generating a HyperContext hyperspace

In a hypertext, the same document can be the destination of many different links. Consequently, the same document may be reached along different paths. It is possible that users who reach the same document following different paths may be looking for different information, or may have reached the same document for different reasons. Such users are likely to interpret the information in the document differently, depending on the other documents in this session the user has so far read and any other knowledge and interests that the user might have. If we are to individualise link and path recommendation knowing only the user's path of traversal through hyperspace, then we need to understand how the information in the child (destination) document is related or relevant to the information in each of the child's parents.

On the Web, web pages range from short and single topic to huge, multi-topic documents. The length of a web page is not a good indicator of the number of topics it is likely to contain. Should information about all topics in a document be added to the short-term user model, in the hope that eventually the dominant topic will float to the surface? Should we use topic distillation algorithms to split up a document into its different topics, and compare each topic to the topic of the region in the parent that the user followed to reach this child? We opted for the second approach to determine the relevant terms in a document visited by the user. A document *interpretation* is a vector of term weights which partially describes a document in the context of a parent. A document has at most $n+1$ interpretations: one for each of its n parents, and an additional one (the *context-free interpretation*), that does not decompose a document into its different topics, which is invoked if a document is accessed directly rather than by following a link to it. To convert the W3C web site to a HyperContext hyperspace we created interpretations for each (HTML) document. A link in the new adaptive hyperspace is retained if the topic distillation algorithm determines that there is sufficient similarity between the topics in the source and destination documents. The user model is updated each time the user traverses a link, using information derived from the visited document's interpretation.

² www.w3.org

3.1 The User Model

The short-term user model is based on the interpretations of documents that the user has accessed during the current session. The user model is used to recommend links each time a document is accessed. A query may also be extracted from the user model and submitted to an information retrieval system to retrieve relevant interpretations if these have been previously indexed.

3.2 Evaluating HyperContext

As we discussed in section 2, our goal is to direct users to relevant information faster than they would be able to find it themselves, particularly when they are unfamiliar with the topic. We describe our original evaluation strategy in section 4. In section 5, we describe the actual strategy we used to evaluate HyperContext. In this paper, we concentrate on the evaluation *process*. The evaluation results are discussed in detail elsewhere [8], [9].

4 Evaluation Strategy 1

The empirical study that we had originally planned was to involve three groups of six participants each. Of the 18 participants, 6 each were previously judged to be novice, intermediate, and advanced information seekers. The initial study involved 36 participants who were set 15 general knowledge information seeking tasks. They were allowed to use any information source (search engine, web directory, their own memory) they liked, but had to indicate if they already knew the answer. For each task, the student had to write down a URL containing the answer (or URLs, if the answer spanned a number of web pages). The information seeking tasks were pre-tested to ensure that the answers were available on the Web.

Each participant's performance for each task was compared to the average time to perform each task (from among those participants who did not already know the answer). Participants who generally arrived at a solution faster than average were considered advanced information seekers, those who were generally much slower at finding information were considered to be novice, and the others were considered intermediate. 6 people were to be randomly selected from each group to participate in the HyperContext evaluation.

A HyperContext Evaluation group was to consist of two novice, two intermediate, and two advanced information seekers. Each group would have an identical set of tasks to perform. The tasks were designed to find technical information, rather than general knowledge as used in the experiment to classify participants. One group would act as the control group, the second and third groups would both use a HyperContext-enabled version of the W3C web site, but the algorithm used to construct the user model would be different. Once again, the performance of the two HyperContext-enabled groups would be compared to the performance of the control group, where we can show that the control group would have used

the equivalent of the W3C web site. Each group would have access to the same information search and retrieval system. The control group would have access to an index generated from the original, unmodified documents, whereas the other groups would have access to an index that also contained an index of document interpretations (document interpretations are discussed in section 3).

4.1 Is a without-adaptation HyperContext equivalent to the Web?

The HyperContext hyperspace created from the W3C web site for use in the evaluation (section 3) can be considered equivalent to the original W3C web site if adaptivity is disabled. By default, the context-free interpretation of a document consists of a vector of term weights for all terms that occur in the document, rather than just those terms that are considered relevant to the parent, when a link is followed. In the disabled version of HyperContext, all link traversals invoke the context-free interpretation, so the interpretation of the document is the same regardless of how the document is accessed. This behaviour is equivalent to the behaviour on the Web. Regardless of how any page is accessed, normally there is absolutely no difference in or about the page that was accessed.

5 Evaluation Strategy 2

Due to a number of unfortunate incidents, including hardware failure resulting in total data loss, and looming deadlines, the intended strategy outlined in section 4 never progressed beyond the first stage of classifying participants as novice, intermediate, and advanced information seekers. By the time the HyperContext hypertext and related data were recovered, there was simply not enough time to re-run the original classification of participants (because their information seeking skills were bound to have changed over time [3]), conduct the rest of the evaluation and analyse the results. Instead, we decided to separate the evaluation of some of the functionality from the evaluation requiring user participation [5]. We developed one completely automated experiment to test our hypothesis about the improved ability to locate relevant information in a HyperContext hyperspace. A second experiment required anonymous Web-based participation from users to judge whether documents recommended by HyperContext were relevant to information they had read on a pre-determined path through a HyperContext hyperspace.

5.1 Locating relevant information

The number of links on a page, coupled with the lack of a link semantics in HTML increases cognitive overhead. A user must decide whether or not to follow a link. Adaptive educational hypertext systems may make use of visual link adaptation to indicate that a link may be followed with profit, or should not yet be followed, e.g., [11]. Alternatively, forms of link hiding [2] may be used, in which users are discouraged from following links unlikely to lead to relevant information. In

either case, this is a form of hypertext partitioning - separating the non-relevant parts of hyperspace from the relevant.

In HyperContext, a user visiting a document actually visits an interpretation of that document. In Section 3 we explained that an interpretation is a vector of term weights, and that different interpretations of the same document may have different vectors of term weights. For instance, in one such vector, some term t_n may have weight w_x . In another interpretation of the same document, the same term may have the same weight, or a completely different weight, depending on how significant the term is to the context of the topic of that document's parent. Interpretations are slightly more complex, however. One interpretation of a document may have link anchors which may or may not be active in other interpretations (a form of link hiding). Additionally, even if the same link is active in several interpretations of the same document, the destination of the link may change depending on the interpretation (figure 1). In this way we are able to partition a HyperContext hyperspace, potentially separating the non-relevant from the relevant.

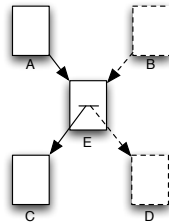


Fig. 1. Link in doc E leads to C if entered from A, and to D if entered from B.

To determine if multiple interpretations of information can adequately partition a hyperspace so that a user can be led to relevant information, we count the number of nodes that must be visited starting from some arbitrary start node until we reach a relevant node. A relevant node is just some randomly selected node that is at least 2 link traversals away from the start node. We compared two adaptive solutions to two non-adaptive solutions, measuring overall performance and the performance of each approach as the path length grew. The adaptive solutions were based on a HyperContext enabled converted W3C hyperspace, and the non-adaptive solutions were based on the original W3C web site. The premise is that the optimal solution is one that finds the shortest path between the start node and the target, relevant, node. The least optimal solution is likely to be based on a breadth-first or depth-first brute force search (depending on the “shape” of the hypertext graph), essentially following the links in the order of least likely to lead to the target node. For this experiment we traversed the links in the order they occurred in a document, using a hybrid approach. We process nodes breadth-first until we encounter the target node. We then prune the graph

of accessed nodes, eliminating all visited nodes to the right of the shortest path between the start node and the target node (figure 2). This is the equivalent of a depth-first search guarded by the known depth of the target node. If the best link to follow always happens to be the first link in a document, then this approach will give results similar to the optimal solution. However, unless the best link is always the last one in a document, then this approach will give better results than the least optimal solution, because nodes which did not need to be visited will not be counted. This approach yielded paths of maximum length 5 (four link traversals from the root).

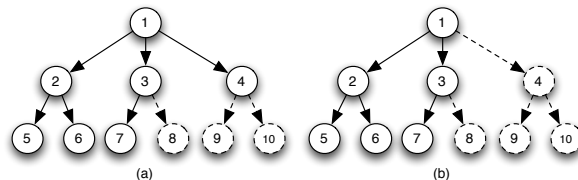


Fig. 2. Node 7 is the target node. (a) Solid nodes are visited in breadth-first search; (b) hybrid depth-first marks node 4 as unvisited.

An algorithm that partitions the hyperspace may decrease the span of the graph, and hence improves the speed with which a target node can be reached, even when a brute-force approach is taken. The efficiency may be decreased if a relevant node is made either unreachable or reachable by a longer traversal of the graph if the hyperspace is partitioned badly (figure 3). In either case (adaptive or non-adaptive) the efficiency may be further improved by introducing a link ordering algorithm that ranks links in a document according to the likelihood that they will lead to the target node. The link ordering algorithm compares the current node’s children (a lookahead of 1) to the target node. Links in the current document are traversed in the order of degree of similarity between the link destination and the target node. In the experiments with the adaptive version of the hypertext, the interpretation of each child (section 3) is used by the algorithm, rather than the context-free interpretation of the child used in the non-adaptive version.

5.2 Evaluating Document Recommendation

In the second part of the evaluation, we prepared a number of paths through hyperspace that all involved exactly four link traversals (for consistency with the maximum path length reported in subsection 5.1) through different documents. If a document was re-visited on a path, the path was not selected for the experiment. Two user models were maintained. We assumed that the first document on the path was the root of the path, and that both user models were empty at this point. Each user model was updated following a link traversal to

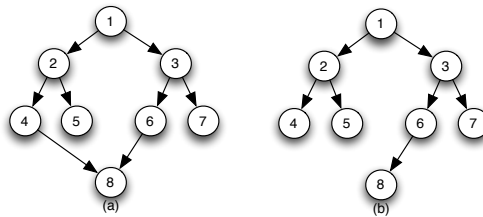


Fig. 3. Partitioned hypertext: (a) before, and (b) after. Node 8 is no longer reachable from node 4 in (b) and so may take longer to reach.

the next document on the path. On reaching the last (fifth) document on the path, two queries were generated, one from each user model, and submitted to our search engine. The first document recommended by each user model was noted. Eventually, participants were asked to give relevance judgements about each recommended document having first read all documents on the path.

A term weight vector based on the interpretation of each visited document on a path is used to update the first user model ($UM_{adaptive}$). For the second user model ($UM_{control}$), the context-free interpretation of the document is used³. If both user models recommended the same document, the path was considered inapplicable for evaluation purposes and was discarded. Eleven conforming paths of length five were randomly selected. The documents on the path and the documents recommended by each user model were placed on-line and hosted by a Web server for 25 days. Members of staff in the Department of Computer Science and AI at the University of Malta and its student population were invited via e-mail to participate in the on-line evaluation. Participation was totally anonymous and could be carried at the participant's leisure from a location of their choice. Participants were asked to read each of the first five documents in a path in the order they were displayed. They were then shown two recommended documents (one after the other) and asked to give a relevance judgement about each.

We used a 4-scale of relevance judgements (highly relevant, quite relevant, quite non-relevant, highly non-relevant), rather than the two (relevant, not relevant) normally used [10], because we expected both user models to make recommendations of at least slightly relevant documents. Participants were not told the order in which recommended documents would be displayed. They did not know which document was recommended by $UM_{adaptive}$ and which was recommended by $UM_{control}$. The sequence was set randomly.

5.3 Summary of Results

The results of the evaluation are reported extensively in [9] and summarised in [8]. To locate relevant information we measured the difference between the best

³ This is the equivalent of the Web version of the document (section 4.1).

case scenario (the shortest path between two nodes), the worst case (the longest path assuming that we know the level depth of the target node), and the adaptive solutions. The adaptive solutions outperformed the non-adaptive ones as path length increased. If the target node was 3 or 4 link traversals from root, then the adaptive solutions found the target node having visited less intermediate nodes than the non-adaptive approaches. This performance was reversed for target nodes that were up to 2 links traversals away from the root.

For the second part of the evaluation, two user models were used to recommend documents to users using an adaptive and a non-adaptive approach respectively. At face value, documents recommended by the non-adaptive approach were considered more relevant than those recommended by the adaptive approach. However, if time spent reading a document is an indication that a document is skim read or read closely (deep read), then readers tended to consider relevant the document recommend by the adaptive approach when the documents were deep read, and those recommended by the non-adaptive approach if the document was skim read. However, this is an assumption because although we measured the amount of time spent reading each document on a path users were not asked to confirm whether they skim or deep read the documents.

6 Conclusion

One main and significant difference between general-purpose adaptive hypertext systems, like HyperContext, and adaptive educational hypertext systems is that our evaluation participants did not necessarily have any motivation to read about or learn about the information contained in our hyperspace (Web standards). In educational hypertexts, there may be more scope for finding participants who are interested in learning what the system is teaching. We feel that HyperContext would have benefited from evaluation by participants who use it to guide their search for information that they are motivated to obtain. However, setting up such experiments can be complex and expensive [4]. For example, the Alberta Ingenuity Centre for Machine Learning pays an honorarium to Web-based participants in the evaluation of LILAC⁴.

Creating test collections with value judgements for adaptive hypertext systems may make the results of automated evaluation more reliable and comparable, as has been the case with information retrieval and systems for some decades [1]. Perhaps the most common criticism of this approach, and one that could also effect adaptive hypertext systems, and not merely because some, like HyperContext, make use of information retrieval systems to make recommendations, is that *relevance* is highly subjective. The Text Retrieval Conference uses “pooling” to set relevance judgements for documents in test collections [10], [7].

We automated some of the evaluation process for HyperContext. We selected the algorithm for updating the user model, and we used a simple topic distillation algorithm to create interpretations of documents based on each of their

⁴ www.web-ic.com/lilac/honorarium.html

parents to partition hyperspace so that we can more quickly locate a target document presumed to contain relevant information. Of course, this automated experiment alone was insufficient to conclude that users would actually find the recommended documents relevant, so we then invited participants to provide relevance judgements for documents that were recommended after the participants had read 5 documents on a path through a converted W3C web site.

We use a short-term user model that is initially empty to collect information about a user's interests as the user navigates through hyperspace. This is the only way in which the user model can be updated. If a user is not permitted to use a search engine to locate information, or to jump directly to pages using their URL, or to directly edit the user model, but can only follow paths through the information space, then the user model of all users following the same path will be updated in the same way, and the same recommendations will be made. If we can also know in advance which links and documents should be recommended at each stage, then it should be possible to create a test collection with relevance judgements that can then be used for automated evaluation.

References

1. Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Chapter 3: Retrieval Evaluation (1999) Addison Wesley:US
2. Brusilovsky, P.: *Methods and techniques of adaptive hypermedia*. *Adaptive Hypertext and Hypermedia* (1998) 1–43. Kluwer Academic Publishers:Netherlands.
3. Chin, D. N.: *Empirical Evaluation of User Models and User-Adapted Systems*. *User Modeling and User-Adapted Interaction*, v. 11, n1-2 (2001) 181–194
4. Del Messier, F., and Ricci, F.: *Understanding Recommender Systems: Experimental Evaluation Challenges*. Weibelzahl, S., and Paramythis, A. (eds) *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems in conjunction with UM2003* (2003) 31–40.
5. Herder, E.: *Utility-Based Evaluation of Adaptive Systems*. S. Weibelzahl and A. Paramythis (eds), *Workshop on Empirical Evaluation of Adaptive Systems User Modeling 2003* (2003) 25–30
6. Höök, K.: *Evaluating the utility and usability of an adaptive hypermedia system*. *Proceedings of the 2nd international conference on Intelligent user interfaces*, Orlando, Florida, United States (1997) 179–186
7. Smeaton, A. F., and Harman, D.: *The TREC Experiments and their impact on Europe*. *Journal of Information Systems* (1997)
8. Staff, C. D.: *The HyperContext framework for Adaptive Hypertext*. *Proceedings of the Thirteenth ACM conference on Hypertext and Hypermedia*, College Park, Maryland, USA (2002) 11–20
9. Staff, C. D.: *HyperContext: A Framework for Adaptive and Adaptable Hypertext* (2001) PhD thesis, University of Sussex.
10. Voorhees, E.: *Overview of TREC 2003*. *Proceedings of the Twelfth Text REtrieval Conference*. (2003)
11. Weber, G. and Brusilovsky, P.: *ELM-ART: An adaptive versatile system for Web-based instruction*. *International Journal of Artificial Intelligence in Education* 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems (2001) 351-384.

Evaluation Experiments and Experience from the Perspective of Interactive Information Retrieval

Ross Wilkinson Mingfang Wu

CSIRO ICT Center,
Melbourne, Australia
{Ross.Wilkinson, Mingfang.Wu}@csiro.au

Abstract. It has long been a tradition of evaluating information retrieval systems with very simple user models and very simple tasks: the task is to retrieve relevant documents to a user need described by a query. TREC, the Text REtrieval Conference sponsored by NIST, raised the bar by providing large scale collections, well defined user needs, independently judged documents, and a specified form of success. Groups from around the world all tackled this same task that allowed wide analysis of just what factors influenced system performance.

Yet there was concern, as system performance improvement did not always lead to human performance improvement, so a concerted effort to study how people interact with information retrieval systems was undertaken in the Interactive Track of TREC. This paper describes this track, some of the experiments that we have undertaken in this track, and highlights some of the real problems in such evaluation.

There are two key issues that we have often observed in interactive information retrieval. The first issue is that human preference is often not correlated with human performance. Consequently, evaluation that relies solely on either form of evaluation is not reliable. The second issue is that genuine improvements are very difficult to validate, as system variation tends to be dominated by task variation and user performance variation. Consequently, the statistical power of these experiments, often already very expensive to conduct because of user participation, can be quite low. Thus we argue for staged experiments where only very “obvious” system performance gains are explored.

In the end, simple performance measures have proved less helpful than deeper analysis of just how people interact with their information systems.

1 From System-oriented Evaluation to User-oriented Evaluation

Information Retrieval (IR) research has long been driven by evaluation. The evaluation was largely on the system part (or so called batch mode evaluation). The basic testing environment is to build a test collection which includes 1) a collection of documents to search on, 2) a set of queries to search for, 3) and the relevance judgement about each document with respect to each query. A testing system is then fed with a set of queries and scored according to its ability to retrieve relevant documents.

In the early stages (of the field), each research groups built their own small test collection (with hundreds of documents and a few queries). As a result, it was hard to compare systems across research groups and generalize the experimental results. The

TREC [7][14] is a major initiative to address these issues. The goal of TREC is to “encourage research in text retrieval based on large test collection”, and to “provide a common task evaluation that allows cross-system comparisons”. Compared to other previous test collections (such as Cranfield test collections [6]), the TREC test collection enlarges greatly the size of document collection, provides more realistic queries, and uses more realistic relevance judgments made by independent assessors. In addition, TREC also provides specialized test collections for different test tasks (tracks), such as: filtering, high precision, question answering and so on.

This kind of evaluation is designed to allow one parameter to be manipulated at a time - this is good for comparison of individual components embedded within systems. The evaluation criteria are usually: precision (the proportion of retrieved documents that are relevant) and recall (the proportion of relevant documents that are retrieved), measuring accuracy and coverage. This evaluation technique enables system testing to tell us what (component and algorithm) works and what doesn't. Such experiments are usually repeatable.

This kind of system-oriented evaluation is essential for creating more effective and efficient systems for retrieving and organising information, although it has its drawbacks. For example, it treats the search as a one off session, the relevance judgement is usually binary and held constant and users are abstracted away from the search loop. In reality, relevance is dynamic, situational, cognitive, and context dependent [5], and users are often involved in an extended search process, especially now that accessing information through the web has become a routine part of life.

Actually since the beginning of the field, there have been active discussions and debates on addressing the issues of the evaluations of IR as an interactive system, of the including users in the evaluation and in the use of IR systems. It has been argued that the evaluation of IR systems includes not only the retrieval effectiveness, but also the utility, satisfaction and use [8] [10][11][12][13]. This is increasingly recognized as we have seen that the effectiveness of a system may not be automatically transferred to the user performance [4] and it is increasingly hard to achieve further improvements in retrieval accuracy.

Since 1995 (TREC4), the TREC started to set up the interactive track. The goal of this track is “to investigate searching as an *interactive* task by examining the *process* as well as the *outcome*” [15]. Like the batch model evaluation, this track is also a common platform for research groups to evaluate their systems, and a forum for researchers in this area to discuss how to evaluate an IR system as an interactive process. Unlike the batch model evaluation, the interactive track includes the user. The user is given a scenario that simulates a real world search problem, and is asked to perform the search and made their own judgement on what information is relevant and what is not. The evaluation criteria display a mixed flavour of IR and HCI, it includes the user performance and accessibility, usability and user satisfaction.

Each year, the track defines a task and develops a set of topics. The topic is a brief statement of the task (or information need). A common test collection is stipulated and a common evaluation criteria and procedure is recommended. The criteria on user performance are usually task dependent, but overall, they are measures of the success a subject in completing a specified task. During the experiment, a set of instruments are used to collect subjects' subjective evaluation of the testing systems. The instru-

ments include an entry questionnaire to collect subjects' demographic information and search experience, a pre-search questionnaire for each topic to get subjects' familiarity with the search topic, a post-search questionnaire for each topic to collect subjects' opinion on the search experience and the perceived task completeness, a post-system questionnaire for each system to collect subjects' use experience of that system, and finally, an exit questionnaire to let subjects compare the two systems using a given set of criteria.

The experiment design concerns three factors: system, topic and subject. The experimental design should be able to measure separately the effect of topics, subjects and systems as well as gather some information about the strength of expected interactions between system and topic, topic and subject, as well as subject and system. Thus a factorial design, Latin-square experiment design, is recommended by the TREC interactive track. Table 1 shows an example of such a design.

As we can see, this kind of design is comparative in nature. Indeed, in the past, the participating groups compared various feedback methods (e.g implicit versus. explicit [2]), answer organizations (e.g. ranked list versus clustering versus visualization [1][9][16]), document summaries (e.g. general summary versus. task-biased summary [17]), the transfer of system performance to the user performance [4] and many more (please refer: <http://trec.nist.gov/pubs.html>).

Table 1. Minimal Latin-square experimental design
(T: test system, C: control system B1 and B2: two blocks of topics)

Subjects	System, Topic	
1	T, B1	C, B2
2	C, B2	T, B1
3	T, B2	C, B1
4	C, B1	T, B2

2 Our Experience

We have been interested in the effect of different delivery methods on the user's performance of a certain task. Through the interactive track platform, we evaluated and compared an organization of clustering search result with a ranked list, and query-biased document summary with generic document summary on the question answering task, and a topic distillation oriented delivery and a general ranked list on the topic distillation task. In this paper, we will just discuss our participation in the question answering task.

The question answering task here is not simply to find a single fact. The task involves collecting multiple independent data items (called instances) from one or more resources and synthesizing them into an answer. Figure 1 shows an example of such a search topic as given to subjects.

The assessment of the user's performance measures the ability of a subject to identify documents that contain topic instances. It is measured by instance recall and

instance precision, here instance recall equates to the proportion of the known topic instances contained in the documents identified by a subject, and instance precision the proportion of the documents identified by a subject that were deemed to contain topic instances. The assessment process, therefore, provides indirect or, more accurately, circumstantial evidence of the effectiveness of the interactive system's ability to help the subject develop an answer to the information need represented by the interactive topic.

<p>Title: Ferry Sinkings</p> <p>Description: Any report of a ferry sinking where 100 or more people lost their lives.</p> <p>Narrative: To be relevant, a document must identify a ferry that has sunk causing the death of 100 or more humans. It must identify the ferry by name of place where the sinking occurred.</p> <p>Detailed of the cause of the sinking would be helpful but are not necessary to be relevant. A reference to a ferry sinking without the number of deaths would not be relevant.</p> <p>Aspects: Please save at least one RELEVANT document that identifies EACH DIFFERENT ferry sinking of the sort described above. If one document discusses several such sinkings, the you need not save other documents that repeat those aspects, since your goal is to identify different sinkings of the sort described above.</p>

Figure 1. An example topic

Investigation 1: Clustering structure versus ranked list

The topics of the above described task are structured. This, when combined with our goal of structuring and organizing information to form 'answers', suggests the hypothesis: that *organizing information with regard to task structure is helpful to users*.

Intuitively, this makes sense. The goal of an interactive subject is to locate documents that pertain to as many different instances of the topic as possible. Given that there is no benefit¹ in locating documents that cover previously discovered topic instances, it would seem desirable to organize the candidate documents in such a way that documents addressing different instances of the query topic were separated into different groups. Ideally, the interactive user could then simply select a single representative document from each instance group. Further, these instance groups could help the user to organize the identified instances into a final answer.

How, therefore, should the candidate information be organized? The approach we chose to explore is clustering. The questions that we attempted to answer include: 1) Can users recognize good clusters? 2) Do users prefer a clustering approach? And 3) Are users more effective with a clustering approach?

We set up two experiments to answer these questions. In the first experiment, we recruited four subjects. Their task was to judge the relevance of a cluster to the topic based only on the description of cluster. As shown in Figure 1, the description of a cluster includes the ten highest-weighted terms from the cluster centroid, five most

¹ In fact, because Interactive Track experiments are conducted within a fixed time limit, it is counter-productive to locate or view documents that only address previously identified instances of a topic.

frequent word phrases from all documents in the cluster, and the title of the top three ranked documents in the cluster. This experiment shows that subjects were able to correctly determine from the cluster description which clusters were likely to contain relevant information, and which were not.

In the second experiment, we aimed to compare the clustering structure (as shown in Figure 3) with a ranked list on the question answering task [16]. This experiment was done in the TREC7 interactive track environment. We recruited sixteen subjects, each of them searched 4 topics on the cluster-based interface, and another 4 topics on the ranked list interface (similar to the design in Table 1). The result showed that overall there is no significant difference in terms of aspectual recall and aspectual precision between the cluster-based interface and the ranked list, although a fairly clear preference for the clustering structure was shown in subjects' comments as captured in questionnaires.

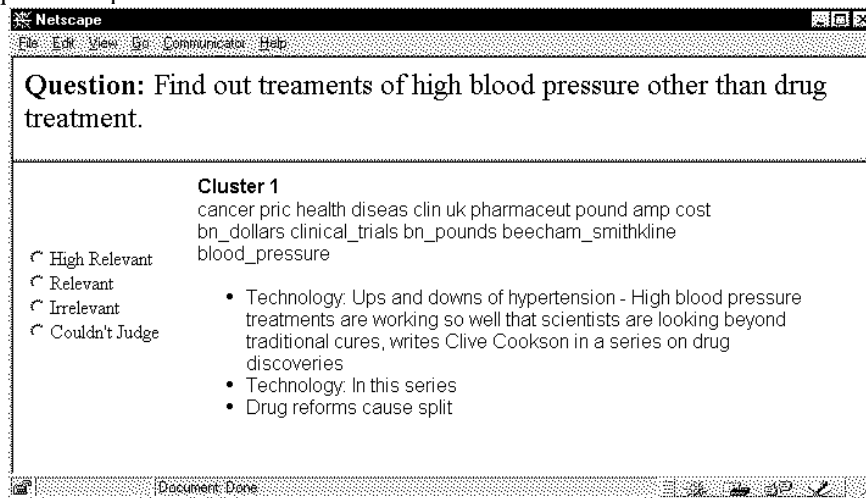


Figure 2. The interface for judging the relevance of clusters

Observations from Investigation 1:

- “Obvious” improvements often do not work. This has often been shown in information retrieval experiments, but extends to work involving users as well. This experiment is an example – there are others we have seen no performance gain.
- There may be a substantial difference between user performance and user preference. We have several times observed that users like a system that provides no performance gain. It is thus important to measure both, to understand the implications of a system variation.

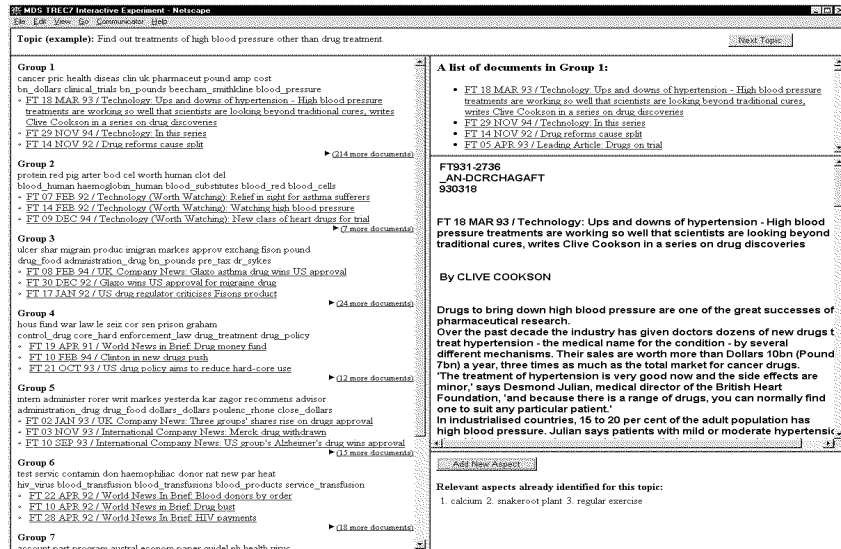


Figure 3. The cluster-based interface

Investigation 2: Task-biased summary versus generic document summary

We analyzed the logged data from the above second experiment, we did find that while the subjects could use the structured delivery format to locate the group of relevant documents, the subjects often either failed to identify a relevant document from the document surrogate or were unable to locate the answer component present in a relevant document. Thus we hypothesized that one of the causes for potential gains from the structured delivery not being realized is that in our test systems the tools used to differentiate the answer containing documents from non-answer containing documents were inadequate for the task of question answering. There is a lack of proper document preview mechanism for subjects to identify the answer-carrying documents.

In the previous experiment, the preview of a retrieved document is represented by its title. The title usually tells the main theme of the document, but very often an answer component (or an instance) exists only within a small chunk of the document, and this small chunk may not necessarily be related to the main theme of the document. For example, for the question “which was the last dynasty of China: Qing or Ming?”, the titles of the first two documents are: “Claim Record Sale for Porcelain Ming Vase” and “Chinese Dish for Calligraphy Brushes Brings Record Price”. The themes of the two documents are “Ming Vase” and “Chinese Dish” respectively, but there are two sentences in each document that mention the time range for Ming Dynasty and Qing Dynasty. By reading only the title, the subjects may miss a chance to find the answer easily and quickly, even the answer components are located in the top ranked documents.

So in our next experiment, we moved on to provide a user a task-biased document preview/summary. The research question we investigated in the experiment is: given a same list of retrieved documents, will the variation in document summaries improve user’s performance on question answering task?

We conducted two experiments that compared two types of candidate lists in two experimental systems [17]. One system uses the document title and the first 20 words of a document as the document’s summary, while the other system uses the document title and the best three “answer indicative sentences” extracted from the documents as the document’s summary (as shown in Figure 4). The first experiment was done in the TREC9 interactive track environment (the documents in the collection are news articles). A second confirming experiment repeated the first experiment, but with different test collection (with web document collection) and subjects. The purpose of the second experiment was to confirm the strong results from the first experiment and to test whether the methodology could be generalized to web data.

Both experiments showed that subjects took less effort (issued fewer queries and read fewer documents), but found significantly more answers from the interface with task-biased document summary than the interface with the general document summary. Subjects also preferred the task-biased document summary. This result indicates that it makes difference by constructing a delivery interface that takes into account the nature of the task.

The set of experiments indicate that different search tasks may require different delivery methods. For the task of finding relevant documents, it has been found that under some circumstances the clustering of retrieved documents was better than a ranked list [1]. However, for the task of question answering, we didn’t find this delivery method performed better than a ranked list. While the second experiment indicates that a relatively simple document summary can significantly improve the searcher’s performance in question answering task.

Observations from Investigation 2:

- In both of these experiments, it appeared intuitively “obvious” that the test system (the clustering structure and the task-biased document summary) offered advantages over the base system (the ranked list). Only in the second case did the evidence back up the obvious.
- In this experiment we found a significant difference in performance. The raw average difference in performance was high – there was a 33% performance difference between the test system and the experimental system. This did give a statistically significant difference, but a smaller experiment, or an experiment where the difference was less extreme might have showed a difference that was not statistically significant. Using an ANOVA analysis below we can see that while the effect of the system is significant, so is the effect of the topic, and to a lesser extent, the user (our user population was chosen to be homogeneous).

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
System	1	1.205128	1.205128	16.54529	0.0001086
User	15	1.870922	0.124728	1.71240	0.0642876
Topic	7	5.234581	0.747797	10.26656	0.0000000
System:User	15	0.820926	0.054728	0.75137	0.7255804
System:Topic	7	0.518039	0.074006	1.01603	0.4262646
Residuals	82	5.972726	0.072838		

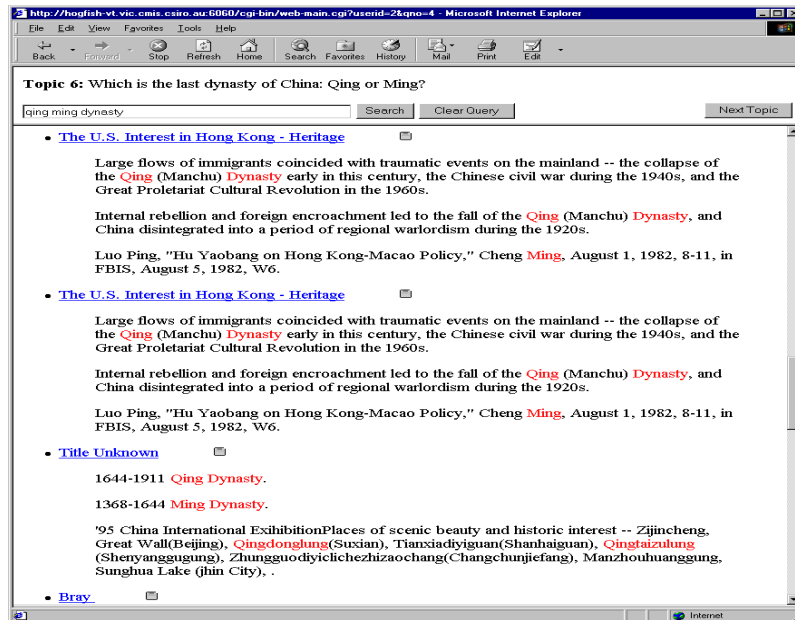


Figure 4. The interface for showing the answer indicative sentences.

3 Lesson Learned

From these experiments, and our observations of other research teams who have contributed to the TREC interactive track, we can make some comments. Firstly from a positive perspective:

- The TREC leverage the effort to build the evaluation platform – there is significant value to research teams working on the same task using the same data. What is learned by one group can be checked with other research teams' outcomes, so it is possible to identify potential trends, but also possible outliers.
- As we know the user involved evaluation is costly, each time only a couple of hypothesis can be tested. A common evaluation platform provides an opportunity for two or more research groups to co-ordinate their experiments in a way so that more hypotheses can be tested, as demonstrated in our current study with Rutger's team [18].
- A carefully constructed study using balanced experimental conditions such as the Latin square design can deal effectively with some of the possible interaction effects – observe the results for the interaction effects in the ANOVA analysis above showing little contribution.
- Effective delivery works in the right context – knowing the detail of the task can allow information to be organised and delivered more effectively than the standard list format of “normal” information retrieval systems

On the other hand:

- Even we had a common evaluation platform, it is hard to make direct comparisons of different systems across sites – there is often just too much variation, and in any case participating research groups were often exploring different hypotheses.
- It is hard to get a statistically significant difference, as the task variation and user variation often overwhelmed potential system variation leading to a lack of experimental power. It is well worth attempting to determine the variation of performance in a small experiment to determine whether there is any hope of achieving a result of statistical significance.
- Repeatability of experiments is difficult and expensive and may well not produce identical/consistent results, particularly given user variation.

It is important to measure both user performance on the task, and user preference. We have observed that these measures often do not correlate (in a short evaluation period and laboratory controlled setting). Yet both are important, if the target user population likes a system refinement, but it does not lead to productivity gains, it is questionable, but equally if performance improves, but the users do not like the approach, then there may well be a problem in uptake. (Maybe a longitudinal study is needed to observe the correlation between preference and the performance.)

4 Conclusions

We have described something of the history of evaluation in information retrieval from the perspective of the TREC interactive track. We then described some of the experiments that we have conducted, and discussed some of the lessons learned.

There are two key issues that we have often observed in interactive information retrieval. The first issue is that human preference is often not correlated with human performance. Consequently, evaluation that relies solely on either form of evaluation is not reliable. The second issue is that genuine improvements are very difficult to validate, as system variation tends to be dominated by task variation and user performance variation. Consequently, the statistical power of these experiments, often already very expensive to conduct because of user participation, can be quite low. Thus we argue for staged experiments where only very “obvious” system performance gains are explored.

In the end, simple performance measures have proved less helpful than deeper analysis of just how people interact with their information systems.

References

1. J. Allan, A. Leuski, R. Swan and D. Byrd. Evaluating combinations of ranked list and visualizations of inter-document similarity. *Information Processing and Management*, Vol. 37, Number 3, pp. 435-458, 2001.

2. N. J. Belkin, C. Cool, D. Kelly, S. J. Lin, S. Y. Park, J. Perez-Carballo and C. Sikora. Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval. *Information Processing and Management*, Vol. 37, Number 3, pp. 403-434, 2001.
3. M. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceeding of the 19th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 76-84, 1996.
4. W., Hersh, A. Turpin, S. Price, D. Kraemer, B. Chan, L. Sacherek, D. Olson, Do batch and user evaluations give the same results? In *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.17-24, 2000.
5. P. Ingwersen. *Information Retrieval Interactions*. Taylor Graham, 1992.
6. K. Sparck-Jones. The cranfield tests. In K. Sparck-djones (editor), *Information Retrieval Experiment*, pp. 256-284. Butterworths, 1981
7. K. Sparck-Jones. Further relections on TREC. *Information Processing and management*, vol. 36, pp.37-85, 2000.
8. F. W. Lancaster. *Information Retrieval Systems: Characteristics, Testing and Evaluation*. New York: John Wiley & Sons, 1979.
9. R. Osdin, L. Ounis and R. W. White. Using hierarchical clustering and summarisation approaches for web retrieval: Glasgow at the TREC 2002 interactive track. In *Proceedings of TREC 2002*.
10. S. E. Robertson and M. Beaulieu. Research and evaluation in information retrieval. *Journal of Documentation*, Vol. 53, Number 1, pp.51-57, 1997.
11. G. Salton. The smart environment for retrieval system evaluation – advantages and problem areas. In Karen Sparck Jones (editor), *Information Retrieval Experiment*, pp. 316-329. Butterworths, 1981.
12. L. T. Su. The relevance of recall and precision in user evaluation. *JASIS*, vol.45, Number 3, pp.207-217, 1994.
13. J. Tague-Sutcliffe. Measuring the informativeness of a retrieval process. In *Proceedings of the 15th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.23-36, 1992.
14. E. M. Voorhees and D. Harman. Overview of the fifth text retrieval conference (TREC-5). In *Proceedings of the fifth text retrieval conference*, pp.1-28, Gainthburg MD,USA, 1997.
15. P. Over. The TREC interactive track: an annotated bibliography. *Information Processing and Management*, Vol. 37, Number 3, pp.369-381, 2001.
16. M. Wu, M. Fuller and R. Wilkinson. Using clustering and classification approaches in interactive retrieval. *Information Processing and Management*, Vol. 37, Number 3, pp. 459-484, 2001.
17. M. Wu, M. Fuller and R. Wilkinson. Search performance in question answering. In *Proceedings of the 24st ACM SIGIR International Conference on Research and Development in Information Retrieval*. pp.375-381, 2001
18. M. Wu, G. Muresan, A. McLean, M. Tang, R. Wilkinson, Y. Li, H. Lee and N. Belkin. Human versus Machine in the Topic Distillation Task. To appear in *Proceedings of the 27st ACM SIGIR International Conference on Research and Development in Information Retrieval*. July 2004.