Vol. 9 No. 2, January 2004

The challenge of automated tutoring in Web-based learning environments for information retrieval instruction

Eero Sormunen & Sami Pennanen
Department of Information Studies
University of Tampere
Finland

change font

Abstract

The need to enhance information literacy education increases demand for effective Web-based learning environments for information retrieval instruction. The paper introduces the Query Performance Analyser, a unique instructional tool for information retrieval learning environments. On top of an information retrieval system and within a given search assignment, the Query Performance Analyser supports learning by instantly visualizing achieved query performance. Although the Query Performance Analyser is a useful tool in training searching skills, performance feedback is not enough for learners practicing alone in Web-based learning environments. The paper reports the findings of a log analysis on user problems in exercising Boolean and best-match queries. A blueprint of an automated tutoring system for IR instruction is presented.

Introduction

The role of the Web as a major Internet service has been characterised by the explosion of information supply including freely accessible Web resources and digital library services. Information retrieval (IR) systems are used not only by information experts, but also by a growing number of other people as a daily routine. Libraries are actively developing programmes and courses to meet the increasing demand for information literacy education. A large share of that work has focused on Web-based learning environments (Dunn 2002, Orr *et al.* 2001).

Query Performance Analyser (QPA) is a tool for the visualization of query effectiveness developed at the University of Tampere. Since 1998, the tool has been used routinely at the University to demonstrate information retrieval phenomena in the classroom and to create search exercises in searching the Web. (Halttunen and Sormunen 2000, Sormunen *et al.* 2002). In 2002, a new version of QPA in Java was launched, and a pilot study to apply the tool in training users of academic and polytechnic libraries was begun.

One of the first lessons of the pilot study was that libraries were lacking resources needed to organize supervised, small group exercises for training students' practical skills in information retrieval. Libraries are forced to count on Web-based course materials and exercises used by students independently at their own pace. This raised a new challenge for the Query Performance Analyser. Our own classroom experience was that students not having the basic skills of searching needed tutoring in exercises to advance in the learning process. The question is: What type of tutoring do the users of QPA need in independently conducted search exercises, and what aspects of the user support could be managed by an automated tutoring system?

The paper starts with a short description of the QPA system and its uses. Next, the results of a log analysis on user errors in search exercises are presented. The paper ends with a discussion on possibilities and forms of automated tutoring in a QPA-supported learning environment.

Query Performance Analyser and information retrieval learning environments

The Query Performance Analyser (QPA) was developed at the Department of Information Studies, University of Tampere, to serve as a tool for rapid query performance analysis, comparison and visualisation (Sormunen *et al.* 1998). On top of a standard test collection, it gives an instant visualisation of performance achieved by any user-generated query in an assigned search task. QPA has been applied both in IR instruction (Halttunen & Sormunen 2000, Halttunen 2003a, 2003b) and in IR research (Sormunen 2000, 2002, Sormunen *et al.*, 2002).

Figure 1 presents the general architecture of QPA (version 5.1). The application consists of two modules. The main application supports three basic functions (query formulation, viewing of results, logout) and six optional functions (change of system settings, performance visualisations, selection of queries and documents into a personal work space, viewing a personal search history, hall of fame, help pages). At present, QPA is interfaced to a traditional Boolean IR system (TRIP) and a probabilistic IR system (InQuery) providing access to the TREC collection and three Finnish test collections.

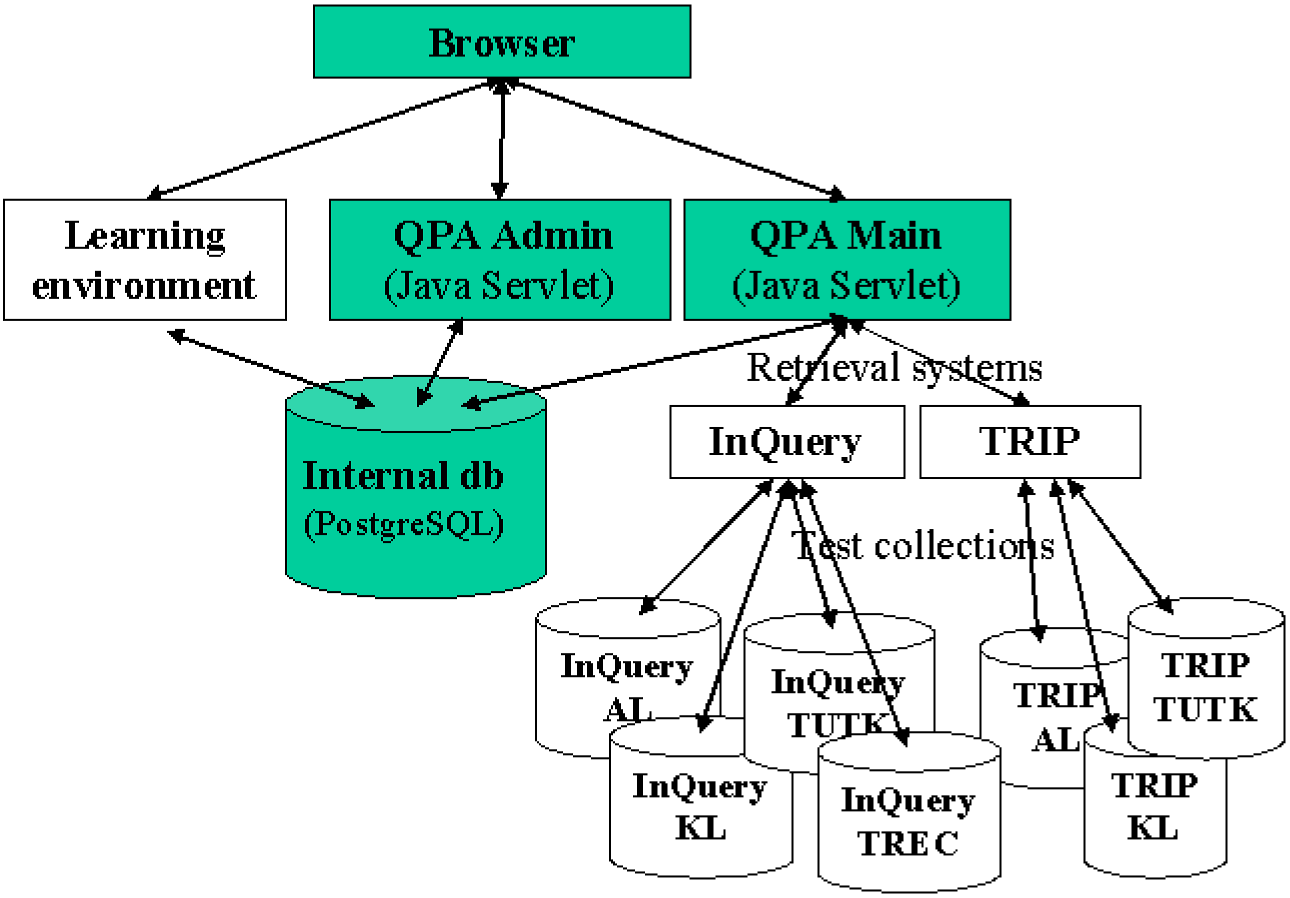
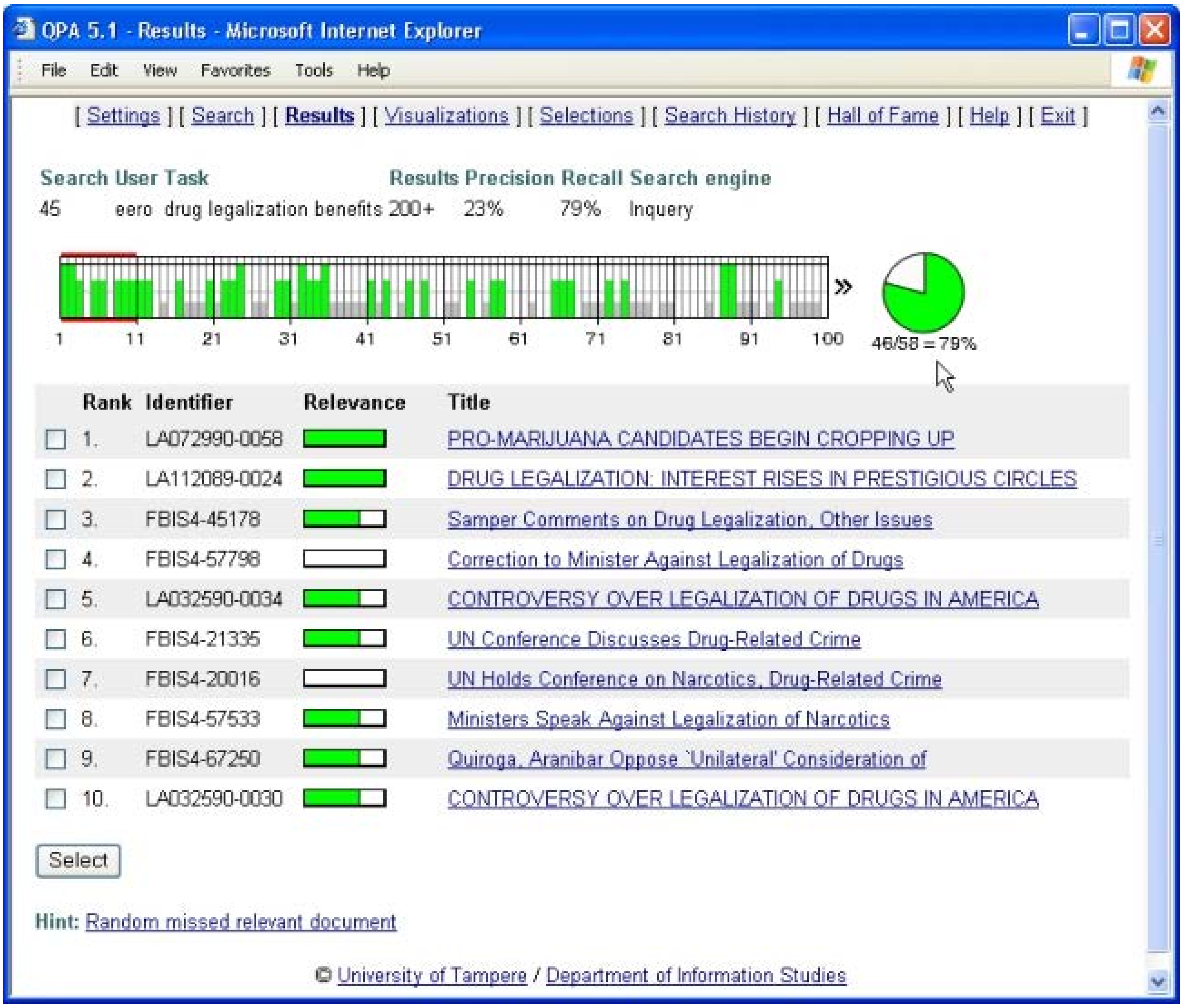


Figure 1: General architecture of the Query Performance Analyser

The Administration module is used to administrate three things: basic search topics, search assignments, and users. Basic search topics are created by composing a search topic description and feeding relevance data. Search assignments (exercises) are created by selecting a search topic, composing the actual assignment text displayed to the user and selecting functions and performance feedback forms available for the learner.

QPA can be linked to any Web-based learning environment using a straightforward procedure. The teacher accesses the Administration module and creates a search assignment on top of a selected search topic. After the search assignment has been created, a link is added on the Web page of a learning environment. By activating the link, the authorised user of the learning environment may access all QPA functions and get all performance feedback that the teacher has selected for this particular exercise.

Figure 2 illustrates the results viewing function of the QPA. In this case, the teacher has created a search assignment on a TREC topic dealing with the potential benefits of drug legalization. All functions and performance feedback are available for the user (not a typical case). The user may observe, for example, from the document bar how many relevant documents were found, and how they are located within the result set. From the recall pie, the user sees how large a share of relevant documents were retrieved (or not retrieved). If the user selects [Visualization], s/he may compare the performance of selected queries in the form of document bars or traditional precision recall curves. We do not discuss further the details of QPA functions here.


QPA 5.1 - Results - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
[Settings] [Search] [Results] [Visualizations] [Selections] [Search History] [Hall of Fame] [Help] [Exit]
1
11
21
31
41
51
61
71
81
91
100
»
46/58 = 79%
Select

Search	User	Task	Results	Precision	Recall	Search engine
45	eero	drug legalization benefits	200+	23%	79%	Inquery

	Rank	Identifier	Relevance	Title
□	1.	LA072990-0058	🟩🟩🟩	PRO-MARIJUANA CANDIDATES BEGIN CROPPING UP
□	2.	LA112089-0024	🟩🟩🟩	DRUG LEGALIZATION: INTEREST RISES IN PRESTIGIOUS CIRCLES
□	3.	FBIS4-45178	🟩🟩⬜	Samper Comments on Drug Legalization, Other Issues
□	4.	FBIS4-57798	⬜⬜⬜	Correction to Minister Against Legalization of Drugs
□	5.	LA032590-0034	🟩🟩⬜	CONTROVERSY OVER LEGALIZATION OF DRUGS IN AMERICA
□	6.	FBIS4-21335	🟩🟩⬜	UN Conference Discusses Drug-Related Crime
□	7.	FBIS4-20016	⬜⬜⬜	UN Holds Conference on Narcotics, Drug-Related Crime
□	8.	FBIS4-57533	🟩🟩⬜	Ministers Speak Against Legalization of Narcotics
□	9.	FBIS4-67250	🟩🟩⬜	Quiroga, Aranibar Oppose `Unilateral' Consideration of
□	10.	LA032590-0030	🟩🟩⬜	CONTROVERSY OVER LEGALIZATION OF DRUGS IN AMERICA

Hint: Random missed relevant document

© University of Tampere / Department of Information Studies

Figure 2: The QPA results viewing function

2.1 QPA as a component of a learning environment

QPA can be used for different purposes: for the tutor in a classroom, QPA is a tool to show the overall effectiveness of any query. It is easy to demonstrate how any reformulation of a query, or any change in the retrieval system reflects on query performance. For the designer of a learning environment, QPA is a tool for creating Web-based searching exercises on which students may work at their own pace. For a student, the analyser is an environment for learning by doing, for example, the query formulation tactics in Boolean or best-match retrieval systems.

In Boolean queries, visualisation of search results with colour coding is an efficient tool to demonstrate the size and the content of the result set. For instance, in a database of news articles it is easy to illustrate how relevant articles occur in clusters in the chronologically ordered result set, sometimes quite far from the top. The observation that so many relevant documents are not retrieved although a query was carefully designed can be a shocking experience for a user who has worked only with traditional searching environments.

In best-match queries, visualisation of search results is very useful when demonstrating the changes in the relevance ranking of documents from one query to another. It is much more difficult for the user to gain control over the search results in best-match systems than in the (exact-match) Boolean systems. Nor is there an established corpus of expertise in everyday best-match searching comparable to that of Boolean searching. Thus, QPA is also an excellent instrument for teachers (and for researchers) to learn, demonstrate, and develop searching strategies for best-match systems.

Empirical studies on the Query Performance Analyser

Halttunen conducted an experimental study on the use of QPA as a component of a learning environment (see Halttunen & Sormunen 2000, Halttunen 2003a and 2003b, Halttunen & Järvelin, in press). From the viewpoint of automated tutoring, the study on scaffolding in information retrieval instruction is especially interesting.

Scaffolding is one of the instructional methods applied in modern learning environments together with modelling, coaching, articulation, reflection and exploration. As defined by Halttunen, scaffolding refers to:

> ...different kinds of supports that learners receive in their interaction with teachers, tutors and different kinds of tools within the learning environment as they develop new skills, concepts or levels of understanding. (Halttunen 2003a: 377)

Scaffolding enables learners to perform activities they were unable to perform without this support. Fading is closely related to scaffolding and represents the idea of gradually removing support when learners can cope with the task independently.

Various types of scaffolds can be applied in information retrieval instruction in both classrooms and instructional tools like QPA. QPA provides the following scaffolds (modified from Halttunen 2003a):

- Giving away parts of a solution: instant query performance feedback in the form of the document bar and recall pie chart.
- Providing cues: give hint text, display an unretrieved, relevant document.
- Providing examples: best queries in the 'Hall of Fame'.
- Providing comparison: comparison of precision:recall curves or document bars of the most recent or selected queries.

In the classroom, the tutor may use other scaffolds:

- Giving away parts of a solution: suggesting query terms.
- Providing cues: suggesting operators, syntax, or query formulation.
- Providing examples: modelling a search process or a search strategy.
- Coaching comments: 'Why did this happen?'
- Asking questions: 'How does that affect...?'
- Providing a timeline: for example, a search process timeline.

When comparing the scaffolds provided by the QPA software and by the human tutor, it is easy to recognize that they complement each other. The key support by QPA is focused on query performance. It is based on 'knowing relevant documents' and on exploiting relevance data in the analysis search results. For the human tutor this type of support is difficult if not impossible. On the other hand, the human tutor can very flexibly help in various types of problems; even in unexpected, occasional incidents. The human tutor may also use the various modes of communication appropriate in particular situations.

Scaffolding and anchored instruction were studied in a quasi-experiment comparing a traditional and an experimental instructional design in an introductory course on IR at the University of Tampere. The course included lectures, independently-made Web-exercises, tutored exercises and course feedback. The experimental group made tutored exercises differently using QPA while the other used operational search systems (Halttunen 2003a).

The main findings of the study were (Halttunen 2003a):

- Students in the experimental group paid more attention to the process approach and the interplay of theory and

practice than students in the traditional group.
- Students in the experimental group spent little less time in weekly exercises and evaluated their learning outcomes better.
- The effectiveness of queries was better in the experimental group.

These findings suggest that the appropriate use of QPA makes IR instruction both more efficient and more effective. On the other hand, the instructional experiment illustrates the limits of scaffolds provided by QPA. If the human tutor is removed from an active role in the instructional setting, a lot of scaffolding functions have to be treated by other means; for example, by learning materials and software-based tutoring.

Methods and data

General environment

We collected log data for this study in the tutored classroom exercises of 'Introduction to Information Retrieval', an introductory course into practical skills of searching in the curriculum of information studies at the University of Tampere. About half of the students had information studies as their major subject and the other half came from other departments of the University. Although the course is an introduction to basic searching skills, some students are not really novices as searchers.

About 120 students attended the course but since, typically, two students were sharing a PC, logs were available for 63 or 64 usernames per exercise. Two search systems were used: a traditional Boolean system, TRIP, and a best-match system, InQuery. A database of some 150,000 news articles covering the years 1997-1999 and downloaded from the local newspaper *Aamulehti* was used in both retrieval systems. All queries were made through the Query Performance Analyser.

Four of the search topics (fish recipes, lifting of a wreck, baseball scandal, and non-military service) provided with comprehensive relevance data available through QPA were used in three exercises for TRIP (fish recipes, lifting of a wreck, and baseball scandal) and in three exercises for InQuery (lifting of a wreck, baseball scandal, and non-military service).

In total, 1,037 queries were recorded for TRIP and 388 for InQuery. We observed two of the seven exercise groups to get a general view of student behaviour and to get some background for the analysis of search logs.

The main differences compared to a real search situation were:

- the search topics were not based on personal needs but given as assignments;
- students worked collaboratively in pairs;
- automatic performance feedback was given by QPA; and
- a human tutor explained assignments and was available for help

Thus, the results should not be generalized to real search situations, although they may reflect some of the problems novice searchers typically have.

Log data samples

We made two types of log analysis. The first data set consisted of low-performance queries not exceeding the level of 10% in average uninterpolated precision (P_{ave}). The sample helps to find errors which have a serious effect on search effectiveness. For this sample of queries, a single primary error for low performance was named applying the three layer model by Borgman (1996) as a framework (see the sub-section 'Error categories' below).

The analysis of low-performance queries does not necessarily help much in revealing user problems in improving non-optimal, partially successful queries. To avoid this problem we made a log analysis in a sample of complete search sessions by two user groups extreme in terms of success. Six search sessions per search task were selected from successful sessions (users called 'high-flyers') and from unsuccessful sessions (users called 'losers') for both TRIP and InQuery. Unfortunately, one of the InQuery search tasks was so trivial that distinct performance differences did not occur, and only two of the best-match search tasks could be used.

In total, 36 search sessions were analysed for TRIP and 24 for InQuery. The same framework by Borgman (1996) was also used here for the analysis of errors but now all errors identified from a query were taken into account. The session-based data helped to identify errors that were solved or remained unsolved during search sessions, and also to find differences between successful and unsuccessful searchers.

Error categories

Borgman's framework contains three layers for the analysis of searcher errors. The layers characterize the level of knowledge with which errors are associated:

- Conceptual knowledge of the information retrieval process—translating an information need into a searchable query;
- Semantic knowledge of how to implement a query in a given system—the how and when to use system features;
- Technical skills in executing the query—basic computing skills and the syntax of entering queries as specific search statements. (Borgman, 1996: 495)

The framework gives a good general basis for a consistent analysis of log data. However, it has to be adapted for the search task and retrieval system at hand and error categories derived inductively from data (see e.g. Sit 1998). In our case, the database contained full-text, indexed news articles, and searching was usually based on plain text contents. In only one of the four search topics, could the user obviously benefit by using field searching (limiting the search to one editorial section of the newspaper).

Since well-specified topics were used in QPA exercises, we used an inclusive query planning procedure developed in an earlier study (Sormunen 2000) to fine-tune the framework. For each topic used in the course, we identified all searchable facets (concepts) defined as an exclusive aspect of a search topic. A facet was regarded as either primary or secondary. A primary facet is a major search concept without which a highly effective query can hardly be formulated. Secondary facets may be used to improve precision but bring in a clear risk of recall losses (in Boolean queries).

For primary facets, we composed comprehensive lists of alternative query terms and tested by QPA which disjunctions ('OR'ed sets) of query terms guaranteed 100% recall. This analysis gave us a framework to analyse which facets or query terms the user might have missed. The opposite case, what facets or query terms used were inappropriate, was easier, since the effect of them could be tested easily by QPA.

Inclusive query plans create a solid ground for error taxonomies, especially at the layer of conceptual knowledge. At the conceptual level, the art of effective searching is based on the searcher's ability to identify the most important aspects of an information need (search topic). Another important question is how the searcher finds appropriate query terms to represent a facet.

We emphasized the analysis of conceptual knowledge errors because most search tasks were totally based on free-text searching. The set of available functions and query elements was small keeping the risk low for errors associated with technical skills and semantic knowledge. The conceptual layer is also more challenging from the viewpoint of automated tutoring.

Results and discussion

Low-performance queries

The results of the error analysis for 334 low performance queries are presented in Table 1. The first, obvious result was that users fail more often in Boolean than in best-match queries. In Boolean queries, 28% of queries (287 out of 1037) performed poorly while in best-match queries only 12% (47 out of 388) lead to drastic performance problems. The low error rate in best-match queries was not a surprise since best-match IR models have been developed to improve system robustness against minor errors in the query fromulation (see e.g., Ingwersen & Willett, 1995).

Category	Boolean queries		Best-match queries		All queries	
	No.	%	No.	%	No.	%
Technical skills errors	**46**	**16**	**40**	**85**	**86**	**26**
Common syntax error	41	14	33	70	74	22
Misspelling	5	2	7	15	12	4
Semantic knowledge errors	**37**	**13**	**7**	**15**	**44**	**13**
Field error	9	3	0	0	9	3
Parenthesis error	14	5	0	0	14	4
AND-operator error	3	1	0	0	3	1
OR-operator error	10	3	0	0	10	3
Proximity operator error	1	0	7	15	8	2
Conceptual knowledge errors	**204**	**71**	**0**	**0**	**204**	**61**
Important facet missed	20	7	0	0	20	6
Special facet missed	41	14	0	0	41	12
Weak facet applied	105	37	0	0	105	31
Important terms missed	17	6	0	0	17	5
Inappropriate terms used	21	7	0	0	21	6
Total	**287**	**100**	**47**	**100**	**334**	**100**

Table 1: Distribution of errors in low performance queries for which average uninterpolated precision was less than 10%. Boolean: 287 out of 1037 queries (28%); Best-match: 47 out of 388 queries (12%).

In best-match queries, most errors were technical, and associated with a special feature of the normalized database index. Best-match queries failed badly also when the user misspelled query terms or applied too restricting proximity operators. (InQuery also supports Boolean and proximity operators contrary to most best-match systems, but their use is not obligatory.) An important result is that conceptual knowledge errors were never the most obvious reason for low effectiveness in best-match queries.

In low-performance Boolean queries, about 70% of errors related to conceptual knowledge. Users applied often weak secondary or inappropriate facets (37% of errors) to limit the size of a result set. On the other hand, users did not always recognize primary facets which led to low precision (primary facets missed 21%). Missed or inappropriately selected query terms were less important but notable categories of errors (13% in total).

Only 13% of errors related to semantic knowledge. Problems were typically associated with the formulation of logically valid query statements. The share of technical errors was slightly higher (16%): these arose mainly from statements lacking operators or having incorrect truncation characters.

The low rate of semantic and technical errors may be explained by the fact that most search tasks required only queries in full-text indexes. However, the high rate of conceptual errors support the earlier findings that Boolean queries are very sensitive to the user's decision on query facets used (see Blair & Maron 1985, Sormunen 2001 and 2002). Any single facet (as a conjunctive block of a Boolean query) may collapse the number of relevant documents retrieved. Either the facet is not explicitly expressed in all relevant documents or the searcher does not discover all terms by which the facet has been referred.

Analysis of search sessions

Performance

Table 2 characterizes queries made by high-flyers and losers in their search sessions. On average, learners formulated 4.9 Boolean queries and 5.4 best-match queries per session. High-flyers stopped earlier than losers (4.4 vs. 5.4 Boolean queries and 5.1 vs. 5.8 best-match queries per session).

Query	High-flyers		Losers		All	
Category	**No.**	**%**	**No.**	**%**	**No.**	**%**
a) Boolean queries by 18 high-flyer and 18 losers.						
Performance OK	59	74	8	8	67	38
Recall problems	15	19	32	33	47	26
Precision problems	6	8	47	48	53	30
Zero hits	0	0	11	11	11	6
Total number of queries	80	100	98	100	178	100
Queries per session	4.4		5.4		4.9	
a) Best-match queries by 8 high-flyers and 8 losers.						
Performance OK	27	66	2	4	29	33
Recall problems	5	12	6	13	11	13
Precision problems	9	22	33	72	42	48
Zero hits	0	0	5	11	5	6
Total number of queries	41	100	46	100	87	100
Queries per session	5.1		5.8		5.4	

Table 2: The number of queries by performance category in search sessions.

In the group of 'high-flyers', 74% of Boolean queries and 66% of best-match queries were effective (average uninterpolated precision P_{ave} close to the best query known). In the group of 'losers', only 8% (Boolean) and 4% (best-match) of queries, respectively, achieved a good level of performance.

'High-flyers' faced more recall than precision problems (19% vs. 8%) in Boolean queries but the situation was the opposite in best-match queries. They did not suffer from the problem of zero hits in either of the systems. The 'loser' group suffered from high rates of precision problems in both systems (48% and 76%). In some cases, they could not find a single document (11% of queries in both systems).

In both systems, more than one half of sessions in the 'losers' group contained multiple types of performance problems (recall/precision/zero hits). The results suggest that 'losers' had serious difficulties in formulating balanced query statements. Performance tended to swing between low recall and low precision.

Errors

The results of the session-based error analysis are presented in Appendix 1 (Boolean) and Appendix 2 (best-match). The total number of 198 errors could be identified from Boolean queries entered in 36 search sessions (about 1.1 errors per query and 5.5 per session). The total number of errors in best-match queries was 109 in 24 sessions (1.3 errors per query and 6.8 per session). Overall differences in error rates were quite small between the IR systems.

Identified errors were categorized into three groups:

A. ***Unsolved errors.*** Errors had a distinct effect on query performance and the users could not solve them during the session. The share of unsolved errors was 26% for Boolean and 17% for best-match queries.
B. ***Solved errors***. Errors had a distinct effect on query performance but the users could solve them consciously or accidentally during the session. The share of solved errors was 40% for Boolean and 24% for best-match queries.
C. ***Errors not really affecting performance.*** These errors were or were not solved but, in any event, did not have a distinct effect on query performance. Thirty-four percent in Boolean and nearly 60% in best-match queries belonged to this category.

In Boolean queries, the distribution between the technical and semantic errors and the conceptual errors is similar to that found in the analysis of separate low performance queries (see Appendix 1 and Table 1). Especially, the similarity of results is obvious in categories A and B having real effect on query performance. The rate of solved and unsolved conceptual knowledge errors is high for Boolean queries. A clear distribution difference can be seen within conceptual knowledge errors when multiple errors per query are taken into account. The role of missed or inappropriately selected query terms becomes more remarkable. The even distribution of conceptual errors into four sub-categories underlines the importance of them all.

In best-match queries (see Appendix 2), a major result was that a very large share of errors (59%) did not have a substantial effect on performance, and this is especially typical for conceptual errors. This is in line with our earlier result that technical and semantic errors are the best single explanation for low performance in best-match queries (see Table 1). However, not all errors dealing with conceptual knowledge are insignificant, on the contrary: the frequency of conceptual errors is well above the level of technical and semantic errors also in categories A and B.

Table 3 shows a summary of how users were able to solve significant errors made during search sessions. In general, 60% of errors in Boolean queries and 68% in best-match queries were solved. Some errors were solved more likely than others. In both query types, technical errors were solved at a high level of probability (82%-88%).

	High-flyers			Losers			All		
Category	**All**	**OK**	**OK%**	**All**	**OK**	**OK%**	**All**	**OK**	**OK%**
a) Boolean queries: 18 high-flyer and 18 loser searchers (see Appendix 1).									
Technical skills errors	2	2	100	15	12	80	17	14	82
Semantic knowledge errors	2	2	100	16	8	50	18	10	56
Conceptual knowledge errors	30	16	53	66	39	59	96	55	57
Important facet missed	6	2	33	18	12	67	24	14	58
Weak facet applied	11	7	64	20	15	75	31	22	71
Important terms missed	9	6	67	18	8	44	27	14	52
Inappropriate terms used	4	1	25	10	4	40	14	5	36
Total	**34**	**20**	**59**	**97**	**59**	**61**	**131**	**79**	**60**
b) Best-match queries: 8 high-flyer and 8 loser searchers (see Appendix 2).									
Technical skills errors	0	0	-	2	0	0	2	0	0
Semantic knowledge errors	0	0	-	14	6	43	14	6	43
Conceptual knowledge errors	11	11	100	18	9	50	29	20	69
Important facet missed	7	7	100	9	4	44	16	11	69
Weak facet applied	0	0	-	3	2	67	3	2	67
Important terms missed	4	4	100	4	2	50	8	6	75
Inappropriate terms used	0	0	-	2	1	50	2	1	50
Total	**11**	**11**	**100**	**34**	**15**	**44**	**45**	**26**	**58**
All = number of errors made (category A and B); Ok = number of solved errors (category B).									

Table 3: The number and percentage of errors made and solved during search sessions.

A surprising result is that, in best-match queries, more technical and semantic errors were made (in proportional sense), and a smaller share of them was solved than in Boolean queries. One might think that the user does not very likely fail in formulating a bag-of-words query. The anomaly was caused by some members of the 'loser' group who tried to apply operators in the InQuery system but failed. InQuery supports multiple operators but their syntax is tortuous. Users were not asked to apply operators but for one reason or another they picked them from the online help linked to QPA's query formulation page. This incident exemplifies how small details of the learning environment may affect the type of errors users tend to make.

In both systems, 'high-flyer' searchers made very few technical and semantic errors that had clear effect on performance and could recover from all of them. The group of 'losers' made quite many technical (14 to 15) and semantic errors (14 to 16). They could solve 80-86% of the technical but only 43-50% of the semantic errors.

'High-flyers' solved about one half of conceptual errors in Boolean queries but all of them in best-match searching. This is an interesting result since it suggests that conceptual errors are more difficult to solve in Boolean than in best-match queries. A potential explanation is that formulating a Boolean query is like walking a tightrope. The searcher has to make a complex decision of including and excluding facets and query terms to balance between recall and precision.

In best-match queries, it is usually advantageous to apply all facets of a topic (not exclude any of them) and then select at least some key query terms for each facet. For empirical findings on optimal Boolean and best-match queries, see (Sormunen 2002). Also Table 3 supports this view. In best-match queries made by 'high-flyers', all conceptual knowledge errors were related to missed facet and terms. The élite of best-match searchers could recover from all conceptual knowledge errors but the best Boolean searchers could solve no more than one half of respective errors .

Discussion

Technical errors (e.g., spelling errors and syntax errors) were quite common but their role seemed to be predominant only for a very small group of users. Unfortunately, this group of users seemed to have problems at all levels of searching knowledge and is obviously a challenge for instruction. In general, technical errors were not critical. Most of the technical errors either did not have a notable influence on performance or could be solved in the course of a search session.

A very typical technical error is that users apply symbols and formulations that are common in Web search engines, for example, the wild-card character ('*') to truncate words. Users also were mixed with two search languages available in the learning environment (Boolean and best-match). Some immaterial database features may have serious consequences in a particular search topic and increase dramatically the risk of technical errors and confuse learners. Exercises should be tested carefully in advance.

Our data on technical errors in best-match queries was biased by one immaterial feature of the database index. Most technical errors were an outgrowth of the word normalization procedure used in the index. All words not in the dictionary of the normalizer are given a special prefix. Searchers were not aware of the feature and suffered from very low performance, since the key query term for that particular topic did not match the same term in the database index. The risk of technical errors is usually associated with the exact-match and complex operators of Boolean search systems. In our data, the unforeseeable behaviour of a single query term exaggerates the role of technical errors in best-match searching.

Semantic errors (typically dealing with the Boolean structure of queries) were a common reason for low performance but in most cases users could manage them. In our case, the semantic environment was not complex since users were mainly making free-text searching in a homogeneous news article database. As with technical errors, advanced searchers made very few and could fix all semantic errors. On the other hand, some users did not have a clear idea how to construct a sound Boolean query. Half of the errors remained unsolved in the group of 'losers'.

Conceptual knowledge errors were dominating in both data sets (low-performance queries versus low or high performance search sessions). Both 'high-flyer's and 'losers' suffered from conceptual errors in Boolean queries but advanced users could solve them in best-match searching. It seems that all users need support at the conceptual level. The challenge is that conceptual decisions are sensitive to subjective judgment and may be based on different search strategies.

The findings seem to be in line with earlier studies on searching errors of novice users. The role of errors associated with conceptual knowledge is typical for various types of information retrieval systems (Borgman 1996, Chen & Dhar 1990, Sit 1998, Sutcliffe *et al.* 2000, Tenopir 1997). The special contribution of our error analysis is that we could use the conceptual structure of inclusive query plans to systematise the analysis of errors. This step is necessarily required in creating an operational framework for an automated tutoring system. Another contribution was that Boolean and best-match queries could be analysed in the same framework.

Towards automated tutoring systems in IR instruction

At the beginning of the paper, we raised two questions:

1. What type of tutoring do the users of QPA need in independently conducted search exercises?
2. What aspects of user support could be managed by an automated tutoring system?

We described first the basic characteristics of the QPA tool and the findings of the scaffolding study by Halttunen (2003b). Then we reported the results of the log analysis on user errors in tutored search exercises. In this section, we try to outline answers to the original questions.

The role and limits of automated tutoring

If we consider tutoring as a form of scaffolding, the unique feature of QPA is that it helps the user to perceive how different moves taken or methods applied in a search session affect retrieval performance. Hints, examples or models can be used in the form of static textual scaffolds also as in any Web-based resource. In learning environments using QPA as a special service in exercises, general and initial support should mainly be given by the learning environment itself. The division of tutorial functions between QPA and the 'mastering' learning environment is an important design issue.

To expand the support provided by the QPA tool, the chances to adopt scaffolds typical for the human tutor have to be considered. Some parts of human support require common-sense knowledge (external to the search exercise) and are out of the scope of automatic tutoring. The analysis of errors made by users presented above shows that users have difficulties at all levels of searching knowledge: conceptual, semantic and technical skills. However, from the learning viewpoint semantic and especially conceptual knowledge is in the heart of tutoring.

The advantage of QPA is that the learner's training situation is limited by the given search assignment and the underlying search topic. It is possible to build a comprehensive representation of the search domain. Building 'intelligent' user help in operational search systems is a much more complex task since searching is based on users' own information needs. (See Bates 1990, Brajnik *et al.*, 2002, Fidel & Efthimiadis 1995, Oakes & Taylor 1998, Shute & Smith 1993, Sormunen 1989). QPA does not treat other access methods than queries in a given database and this reduces also the complexity of the tutoring task. The task is by no means a trivial one but can be regarded as manageable.

Intelligent tutoring systems as a model

Intelligent tutoring systems (ITS) are a traditional knowledge engineering approach applicable here. ITS are intended to give feedback and guidance to a learner in a way similar to a human tutor (Hume *et al.*, 1996, Khuwaja & Patel 1996, Shute & Psotka, 1996). Three main components of a typical ITS implementation are:

- The domain model (or curriculum). This model represents the domain of knowledge that the student is studying.
- The student model. The system attempts to assess what the student already knows.
- The tutor (or teaching strategy). The module is planning which curriculum element ought to be instructed next and how it should be presented to the learner.

The domain model of in the QPA tutoring system could be, for instance, a rule-base representing general query formulation heuristics relevant in the IR setting used and a table of facts about the search topic used. Some heuristics are general (for example, the user needs to identify at least the key facets of a topic and at least some of the query terms for each facet), some are dependent on the system used (the user applies only key facets in Boolean queries but most, if not all, facets in best-match queries) and some are search task dependent (the user applies special facets, e.g. author names). By topic dependent facts we mean, for example, lists of primary and secondary facets and corresponding query term candidates. Descriptive data may be assigned both to facets and query terms.

By applying searching heuristics and topic-related facts, the errors and weaknesses of any query can be identified and let the user know them. Unfortunately, this is not enough since mechanical reporting of faults and giving direct advice to solve those does not necessarily lead to desired learning outcomes. The role of the tutor (in the ITS model) is to select an instructional strategy and generate an appropriate response to the user. This is not a trivial task since QPA is sharing the responsibility for scaffolding with the 'mastering' learning environment. A pragmatic strategy is that QPA provides a defined set of scaffolds and the designer of the learning environment generates complementary support for the user (for example, initial scaffolds).

The ultimate goal of the student model is to build and update a representation of the student's current conception of the domain. Research on ITS has not managed to develop student models which work well in practice (Shute & Psotka, 1996). In the case of QPA, it might be more productive to focus more on the analysis of user errors made in the search session or across search sessions. The analysis of query logs gave us examples of the types of errors the so-called 'loser' group tended to make. The error profile of the user can be used in selecting the appropriate level in giving hints or other scaffolds to the user.

A tentative blueprint for an automated tutoring system

We have started to work on a prototype of an automated tutoring system for the QPA tool. Our main ideas are the following:

1. To begin with, we focus on errors at the level of conceptual knowledge (but do not neglect the levels of semantic knowledge and technical skills). The basic conceptions of information retrieval are associated with the conceptual level. Conceptual knowledge of searching is generic and can be applied in different retrieval environments.
2. A comprehensive model of the search topic based on an inclusive query plan is in the core of the tutoring system. Without this model of the topic (actually data in the form of a table) situation-dependent user support is not possible at the conceptual level.
3. General searching heuristics (e.g., in the form of rules) exploit data on the search topic and on queries made by the user to identify all potential errors and weigh their significance.
4. A separate problem is to develop a procedure how and in what form user feedback is given. The user error profile has here a key role. Also the designer of the learning environment should have a chance to pre-set what options of user feedback are applied.

At the time of writing (December 2003) the first prototype has been completed and the first laboratory tests are underway. The next step is to correct main bugs and functional deficiencies. User tests are scheduled for the second and third quarter of 2004.

Acknowledgements

The work on the automated tutoring system for QPA has been supported by the University of Tampere and by the Ministry of Education. The authors thank Kai Halttunen for his valuable advice and cooperation in the course of this project. We are grateful to the editors for their help in improving the final text.

The InQuery search engine was provided by the Center for Intelligent Information Retrieval at the University of Massachusetts and the TRIP search engine by TietoEnator, Inc., Finland.

References

- Bates, M.J. (1990). Where should the person stop and the information search interface start? *Information Processing & Management* **26**(5), 575-591.
- Blair, D.C. & Maron, M.E. (1985). An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the ACM* **28**(3), 289-299.
- Borgman, C.L. (1996). Why are online catalogues still hard to use? *Journal of the American Society for Information Science*, **47**(7), 493-503.
- Brajnik, G., Mizzaro, S. & Tasso, C. (2002). Strategic help in user interfaces for information retrieval. *Journal of the American Society for Information Science and Technology* **53**(5), 343-358.

- Chen, H. & Dhar, V. (1990). User misconceptions of information retrieval systems. *International Journal of Man-Machine Studies* **32**(6), 673-692.
- Dunn, K. (2002). Assessing information literacy skills in the California State University: a progress report. *The Journal of Academic Librarianship* **28**(1), 26-35.
- Fidel, R. & Efthimiadis, E.N. (1995). Terminological knowledge structure for intermediary expert systems. *Information Processing & Management* **31**(1), 15-27.
- Halttunen, K. (2003a). Scaffolding performance in IR instruction : exploring learning experiences and performance in two learning environments. *Journal of Information Science* **29**(5), 375-390.
- Halttunen, K. (2003b). Students' conceptions of information retrieval : implications for design of learning environments. *Library and Information Science Research* **25**(3), 307-332.
- Halttunen, K. & Järvelin, K. (in press). Assessing learning outcomes in two information retrieval learning environments: a design experiment. Submitted for publication in *Information Processing & Management*.
- Halttunen, K. & Sormunen, E. (2000). Learning information retrieval through an educational game. *Education for Information* **18**(4), 289-311. Retrieved 4 July, 2003 from http://www.info.uta.fi/tutkimus/fire/archive/EfI1820004.pdf
- Hume, G., Michael, J., Rovick, A. & Evens, M. (1996). The use of hints and computer tutors: the consequences of the tutoring protocol. In *Proceedings of the 2nd International Conference on the Learning Sciences*, (pp. 135-142). Evanston, IL, Northwestern University. Retrieved 4 July, 2003 from http://www.valpo.edu/home/faculty/ghume/icls.ps
- Ingwersen, P. & Willett, P. (1995). An introduction to algorithmic and cognitive approaches for information retrieval. *Libri* **45**(3/4), 160-177.
- Khuwaja, R. & Patel, V. (1996). A model of tutoring based on the behaviour of effective human tutors. In Claude Frasson, Gilles Gauthier, Alan Lesgold (Eds.). *Proceedings of the Intelligent Tutoring Systems Conference (ITS '96).* (pp. 130-138). Heidelberg: Springer-Verlag. (Lecture Notes in Computer Science 1086)
- Oakes, M.P. & Taylor, M.J. (1998). Automated assistance in formulation of search statements for bibliographic databases. *Information Processing & Management* **34**(6), 645-668.
- Orr, D., Appleton, M. & Wallin, M. (2001). Information literacy and flexible delivery: creating a conceptual framework and model. *The Journal of Academic Librarianship*, **27**(6), 457-463.
- Shute, V.J. & Psotka, J. (1996). Intelligent tutoring systems: past, present, and future. In David Jonassen (Ed.), *The Handbook of Research for Educational Communications and Technology*. (pp. 570-600). New York: Macmillan.
- Shute, S.J. & Smith, P.J. (1993). Knowledge-based search tactics. *Information Processing & Management* **29**(1), 29-45.
- Sit, R.A. (1998). Online library catalog search performance by older adult users. *Library & Information Science Research* **20**(2), 115-131.
- Sormunen, E. (1989). A knowledge base for the search profile analysis and user guidance. In *Proceedings of the 13th International Online Information Meeting*, London 12-14 December 1989. (pp. 435-446). Oxford: Learned Information.
- Sormunen, E. (2000). *A method for measuring wide range performance of Boolean queries in full-text databases*. Doctoral Thesis. Tampere: University of Tampere. (Acta Electronica Universitatis Tamperensis 748). Retrieved 4 July, 2003 from http://acta.uta.fi/pdf/951-44-4732-8.pdf
- Sormunen, E. (2001). Extensions to the STAIRS study - empirical evidence for the hypothesised ineffectiveness of Boolean queries in large full-text databases. *Information Retrieval*, **4**(3/4), 257-274. Retrieved 4 July, 2003 from http://www.info.uta.fi/tutkimus/fire/archive/INRT87-JKwithFigs1.pdf
- Sormunen, E. (2002). A retrospective evaluation method for exact-match and best-match queries applying an interactive query performance analyser. In Fabio Crestani, Mark Girolami, C.J. van Rijsbergen (Eds.), *Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research Glasgow, UK, March 25-27, 2002 Proceedings*. (pp. 334-352). Heidelberg: Springer-Verlag. (Lecture Notes in Computer Science 2291) Retrieved 4 July, 2003 from http://www.info.uta.fi/tutkimus/fire/archive/ES_ecir.pdf
- Sormunen, E., Halttunen, K. and Keskustalo, H. (2002). *Query performance analyser - a tool for bridging information retrieval research and education.* Tampere: University of Tampere, Department of Information Studies. (Research Notes 2002-1). Retrieved 4 July, 2003 from http://www.info.uta.fi/julkaisut/pdf/qparn1.pdf
- Sormunen E., Laaksonen J., Keskustalo H., Kekäläinen J., Kemppainen H., Laitinen H., Pirkola, A. and Järvelin K. (1998). The IR game - a tool for rapid query analysis in cross-language IR experiments. In *PRICAI '98 Workshop on Cross Language Issues in Artificial Intelligence*. Singapore, Nov 22-24 November, 1998. (pp. 22-32). Singapore: Pacific Rim International Conference on Artificial Intelligence.

- Sutcliffe, A.G., Ennis, M. & Watkinson, S.J. (2000). Empirical studies of end-user information searching. *Journal of the American Society for Information Science and Technology* **51**(13), 1211-1231.
- Tenopir, C. (1997). Common end user errors. *Library Journal*, **122**(8), 31-32.

Appendix 1.

Errors made and solved by 'high-flyers' and 'losers' in Boolean queries. Log analysis of 18 search sessions per group.

	High-flyers		'Losers'		All	
Category	**No.**	**%**	**No.**	**%**	**No.**	**%**
A) Errors unsolved						
Technical skills errors	0	0	3	8	3	6
Semantic knowledge errors	0	0	8	21	8	15
Conceptual knowledge errors	14	100	27	71	41	79
Important facet missed	4	29	6	16	10	19
Weak facet applied	4	29	5	13	9	17
Important terms missed	3	21	10	26	13	25
Inappropriate terms used	3	21	6	16	9	17
Total (unsolved)	*14*	*100*	*38*	*100*	*52*	*100*
Percentage of grand total		*23*		*28*		*26*
B) Errors solved						
Technical skills errors	2	10	12	20	14	18
Semantic knowledge errors	2	10	8	14	10	13
Conceptual knowledge errors	16	80	39	66	55	70
Important facet missed	2	10	12	20	14	18
Weak facet applied	7	35	15	25	22	28
Important terms missed	6	30	8	14	14	18
Inappropriate terms used	1	5	4	7	5	6
Total (solved)	*20*	*100*	*59*	*100*	*79*	*100*
Percentage of grand total		*33*		*43*		*40*
C) Errors not really affecting performance						
Technical skills errors	8	31	28	68	36	54
Semantic knowledge errors	6	23	3	7	9	13
Conceptual knowledge errors	12	46	10	24	22	33
Important facet missed	0	0	0	0	0	0
Weak facet applied	2	8	2	5	4	6
Important terms missed	10	38	7	17	17	25
Inappropriate terms used	0	0	1	2	1	1
Total (no effect)	26	100	41	100	67	100
Percentage of grand total		43		30		34

All errors						
Technical skills errors	10	17	43	31	53	27
Semantic knowledge errors	8	13	19	14	27	14
Conceptual knowledge errors	42	70	76	55	118	60
Important facet missed	6	10	18	13	24	12
Weak facet applied	13	22	22	16	35	18
Important terms missed	19	32	25	18	44	22
Inappropriate terms used	4	7	11	8	15	8
Grand total	60	100	138	100	198	100

Appendix 2.

Errors made and solved by 'high-flyer' and 'losers' in best-match queries. Log analysis of 18 search sessions per group.

	'High-flyers' (18)		'Losers' (18)		All (36)	
Category	**No.**	**%**	**No.**	**%**	**No.**	**%**
A) Errors unsolved						
Technical skills errors	0	-	2	11	2	11
Semantic knowledge errors	0	-	8	42	8	42
Conceptual knowledge errors	0	-	9	47	9	47
Important facet missed	0	-	5	26	5	26
Weak facet applied	0	-	1	5	1	5
Important terms missed	0	-	2	11	2	11
Inappropriate terms used	0	-	1	5	1	5
Total (unsolved)	*0*	*-*	*19*	*100*	*19*	*100*
Percentage of grand total		*0*		*31*		*17*
B) Errors solved						
Technical skills errors	0	0	0	0	0	0
Semantic knowledge errors	0	0	6	40	6	23
Conceptual knowledge errors	11	100	9	60	20	77
Important facet missed	7	64	4	27	11	42
Weak facet applied	0	0	2	13	2	8
Important terms missed	4	36	2	13	6	23
Inappropriate terms used	0	0	1	7	1	4
Total (solved)	*11*	*100*	*15*	*100*	*26*	*100*
Percentage of grand total		*23*		*25*		*24*
C) Errors not really affecting performance						
Technical skills errors	0	0	2	7	2	3
Semantic knowledge errors	6	16	0	0	6	9
Conceptual knowledge errors	31	84	25	93	56	88

Important facet missed	8	22	12	44	20	31
Weak facet applied	8	22	1	4	9	14
Important terms missed	15	41	12	44	27	42
Inappropriate terms used	0	0	0	0	0	0
Total (no effect)	*37*	*100*	*27*	*100*	*64*	*100*
Percentage of grand total		*77*		*44*		*59*
All errors						
Technical skills errors	0	0	4	7	4	4
Semantic knowledge errors	6	13	14	23	20	18
Conceptual knowledge errors	42	88	43	70	85	78
Important facet missed	15	31	21	34	36	33
Weak facet applied	8	17	4	7	12	11
Important terms missed	19	40	16	26	35	32
Inappropriate terms used	0	0	2	3	2	2
Grand total	*48*	*100*	*61*	*100*	*109*	*100*

Scholar Search

Find other papers on this subject.

Google Search

How to cite this paper:

Sormunen, E & Pennanen, S. (2004) The challenge of automated tutoring in Web-based learning environments for IR instruction. *Information Research*, **9**(2) paper 169 [Available at http://InformationR.net/ir/9-2/paper169.html]

Check for citations, using Google Scholar

6026
Web Counter

Last updated: 11 December, 2003