Query ReFormulation on the Internet: Empirical Data and the Hyperindex Search Engine

P.D. Bruza
School of Information Systems[1]
Queensland University of Technology, Australia
bruza@icis.qut.edu.au

S. Dennis
Department of Psychology[1]
University of Queensland, Australia
mav@psy.uq.edu.au

Abstract

Often queries to internet search engines consist of one or two terms. As a consequence, the effectiveness of the retrieval suffers. This paper describes an internet search engine that helps the user formulate their query by a process of navigation through a structured, automatically constructed, information space called a hyperindex. In the first part of this paper, the logs of an internet search engine were analyzed to determine the proportions with which different types of query transformation occur. It was found that the primary transformation type was repetition of the previous query. Users also substitute, add and delete terms from a previous query and with lower frequency split compound terms, make changes to spelling, punctuation, and case and use derivative forms of words and abbreviations. The second part of the paper details the hyperindex - which aids the user in query term addition and deletion. The architecture of a hyperindex-based internet search engine is presented. Some initial practical experiences are also discussed.

Keywords:

internet searching, hyperindex, query formulation

[1]Also: Research Discovery Unit, Research Data Network Cooperative Research Centre, Level 7, Gehrmann Laboratories, The University of Queensland, Brisbane, 4072 Australia, http://www.dstc.edu.au/RDU/

1 Introduction

Users of internet search engines typically engage in a process of query formulation and reformulation in order to fulfill their information needs. An initial query is given and then updated in the light of the results obtained until the user is satisfied that they have a relevant set of documents or that no such set exists. There are at least two reasons why query reformulation occurs.

Firstly, the user may have a quite specific information need in mind but is uncertain how to express that need in the query language. Even when the syntax of the query language is readily comprehensible, as in key word searching mechanisms, it is still necessary to determine which of a set of semantically similar terms is most appropriate for the current query given a particular engine. It is also often necessary for a user to exclude documents by specifying a term which is not related to their information need. For instance, having used the term "surfing" to query for wave surfing documents a user might update their query by excluding "internet" surfing documents.

Secondly, the user's information need may alter as a consequence of examining the search results. For instance, a user starting with the query "quilting" might observe a number of quilting stores and then decide to update their information need to quilting stores in their location. The process of information need update is likely to be particularly prevalent in general purpose heterogeneous domains such as the internet.

In this paper, we will first provide empirical data on the nature of the query reformulation process focusing on the types transitions users make between successive queries and then present the Hyperindex search engine an experimental internet-based information retrieval mechanism designed to:

1. provide support for query formulation by allowing the user to navigate in a structured way through a hypertext of search terms
2. decrease information overload when browsing search results

2 How do users update their queries?

When users are trying to locate information they are typically uncertain both about the precise information they require and the appropriate way to express their information need in the query language of the search engine. To develop computer aided query formulation support systems it is useful to consider the operations that people undertake during the specification process when unaided. The search sequences used in the process of specifying queries provide insight into the strategies users employ to revise their queries in the quest for more specific information.

This section of the paper analyses the logs of the baby Open Information Locator (baby-OIL) a prototype developed by the Resource Discovery Unit of the Research Data Network Cooperative Research Centre. The baby Open Information Locator (babyOIL) accesses information from seven remote information source types including WWW pages, library catalogues, technical report databases, software databases, people and organisation indexes, online document databases and image databases. In addition, babyOIL provides access to

Key	Transformation	Examples
SPL	Term splitting or joining	rockclimb → rock climb joining pals → penpals centre point → centrepoint
DEL	Term deletion - a term is deleted to make the query less specific	malaysia electricity → malaysia
ADD	Term Addition - a term is added to make the query more specific	windows95 → windows95 help
REP	Repetition of query that has already been used	
SUB	Term substitution i.e. substitution of semantically related terms. Instances in which there is both substitution and addition or subtraction are classified as SUB	electronic commerce → electronic contract ezekiel.wav → ezekiel.au
DER	Derived forms of words	jobs → job tourism → tour
SPE	Spelling corrections	
ABR	Abbreviation expansion or contraction	jpl → jet propulsion laboratories
PUN	Punctuation change such as hyphenation removal and addition	hitch-hikers guide → hitchhikers guide
CAS	Case changing	
MIS	Miscellaneous - didn't fit in other categories	

Figure 1: *Query Transformation Types*

the University of Queensland library and phone directory. The interface is available to all members of the internet community [1] .

2.1 Query Transformation Statistics

The analysis of query transformation statistics was restricted to web queries as it was assumed that they would show the greatest variety of mechanisms. The queries were recorded in the log in chronological order without tagging the identity of the person issuing the query. Consequently, temporal proximity in conjunction with semantic relatedness were used to establish which queries belonged to the same search. Each query was considered in turn and related to the preceding queries. In the majority of cases it was clear which queries were part of the same search, however, some queries were being conducted over very long periods of time. For instance, a query for `http://www.ozemail.com.au/~rellis/` occurred on Mon May 13 05:32:10 1996 and again on Tue May 14 06:03:05 1996, more than twenty four hours later. The queries are probably part of the same search, but it is impossible to know with certainty. On occasions this problem was exacerbated by a slight change in topic (i.e. a semantic substitution). Also there were unfamiliar terms which may have been semantically related, in the mind of the user, but which were not classified as such. Because of these factors, it is likely the results represent an underestimation of actual figures.

Of the 4064 web queries that occurred, 1040 were categorised manually into one of 11 query transformation types (see table 1) [2] .

[1] It is available at http://www.dstc.edu.au/projects/babyOIL/

[2] The tagged corpus is available at http://psy.uq.edu.au/ mav/SearchEngineUsage/WebQueries

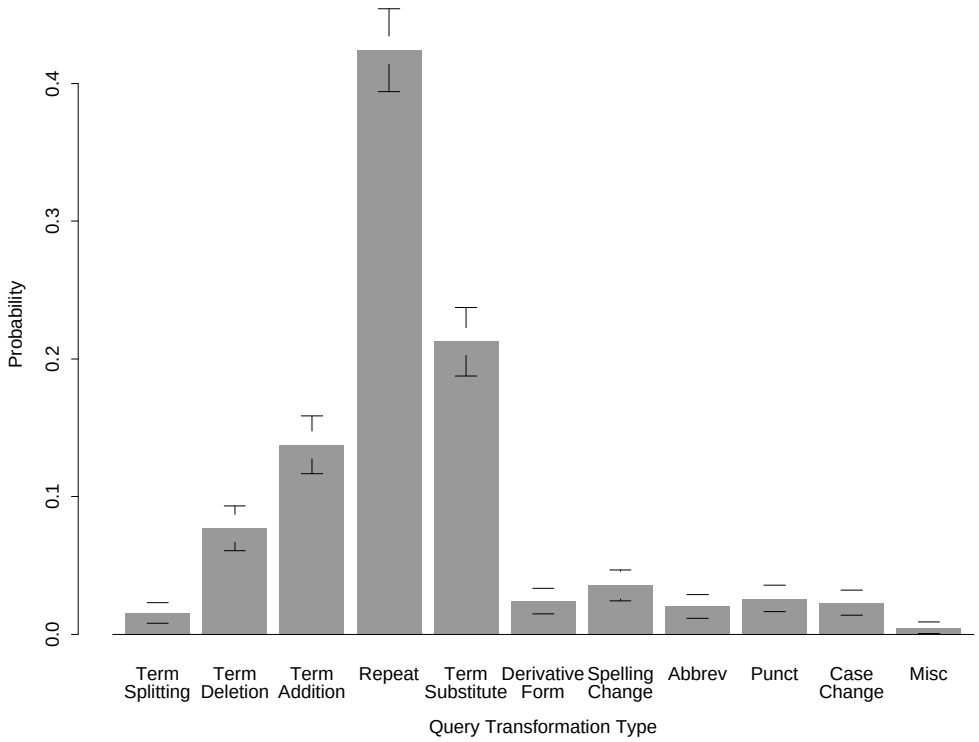

Figure 2: *Query Transformation Type Probabilities*

Figure 2 depicts the breakdown of probability by Query Transformation Type. Bars represent the 95% confidence intervals on the estimation. These were calculated by considering the sample as a series of Bernoulli trials with respect to each of the information types. The variance of the frequency is $(1-p)pn$ were p is the probability of the category and n the number of observations. The standard deviation of the category probability is $\sqrt{p(1-p)/n}$ and the confidence interval half width is calculated by multiplying by $z(0.025)$.

The dominant feature is the large number of times that users repeat a query that they have already issued. This suggests that appropriate caching will improve performance markedly. The other main classes are term substitutions, additions and deletions in that order. The other transformation types showed smaller rates of usage, but all (except miscellaneous) were significantly greater than zero and hence should be incorporated into the query refinement process.

The results of the corpus analysis suggest that query formulation aids should concentrate on mechanisms for predicting which terms are likely to be added to, deleted from or substituted into the query. For instance, terms could be added on the basis of association with current terms (in a thesaurus type mechanism), on the basis of co-occurrence in articles, or on the basis of co-occurrence in user queries. When classifying the log queries, however, it was noted that users tend to prefer linguistically well formed strings. So for instance, in the example of term addition above, "windows95" is converted to "windows95 help". The term "windows95" is converted to an adjective and "help" is added as a noun making the query linguistically well formed. Similarly, in the example of substitution above "electronic commerce" becomes "electronic contract" both of which are linguistically well formed. These linguistic fragments

are often more specific in meaning than strings of associatively related terms and may be particularly important when query length is small.

The next section outlines the hyperindex search engine, a mechanism that makes predictions that preserve the linguistic wellformedness of queries by doing a simple grammatical analysis of retrieved document titles.

3 Hyperindex Search Engine

The hyperindex search engine is designed specifically to:

1. help the user home-in onto a precise description of their information need
2. reduce information overload by presenting the search result at a higher level of abstraction (thus relieving the user of much of the confusing detail encountered when browsing search results).

In order to provide the intuition behind a hyperindex search engine a small example is presented. For more details about hyperindices the reader is referred to [Bru93, Bru90, BBB91, BW92].

Assume that the user has issued the query internet to the hyperindex search engine. This query is then passed on to one or more search engines which evaluate the query. The result sets are merged and passed back to the hyperindex search engine and a hyperindex is created. The user interface to the hyperindex is depicted in the top of figure 3 with a fragment of the underlying hyperindex shown in the lower part of the figure.

Essentially, the user browses over the hyperindex starting from ϵ, which corresponds to an information need which would be satisfied by all documents. In our example, the user made the first step towards refining their given information need by entering the keyword internet. This is the current focus of the search. The descriptors surrounding the focus give possibilities to *refine* the focus, i.e., make it more specific, or to *enlarge* it, i.e., make it more general. In the depictions of the user interface, refinements are denoted by $\triangle$ and enlargements by $\bigtriangledown$.

Refinements signify descriptors which are more specific than the focus. Moreover, they also provide clues regarding various contexts based on the focus. This contextual information is particularly useful for the user whose information need is not clear. In the example, the user chooses to refine the focus by activating internet security resulting in a new screen (see top of figure 4). In other words, the user is expressing that relevant internet documents are those about security. In general, navigation paths through the hyperindex can involve enlargements as well as refinements. For example, imagine the user has refined internet into internet security into internet security software. At this stage the user may decide that internet security software is a too specific description of their information need. In the hyperindex, the user can enlarge internet security software into either internet security or security software. (See the fragment of the hyperindex depicted in figure 5).

The above example demonstrates how the user can navigate through the hyperindex to home-in on a description of their information need. Once a satisfactory description is found,

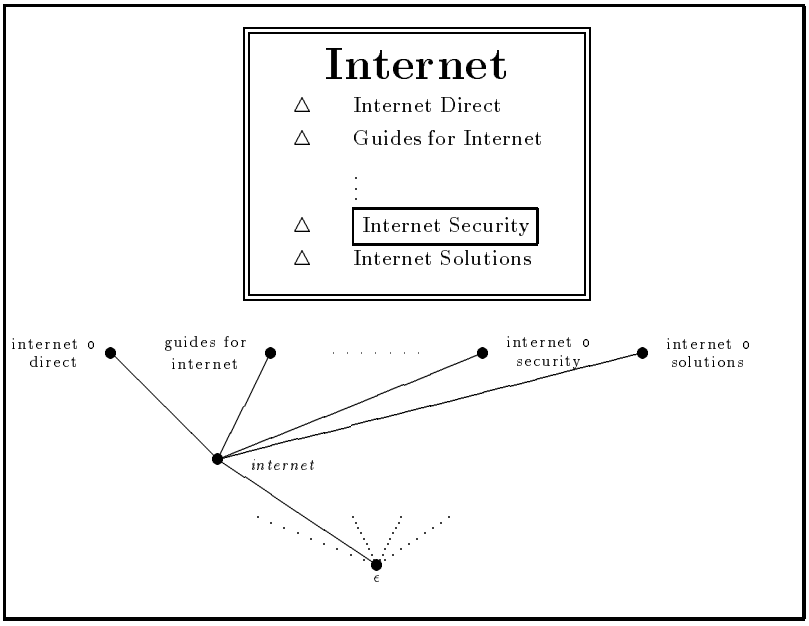

Figure 3: *Before refining*

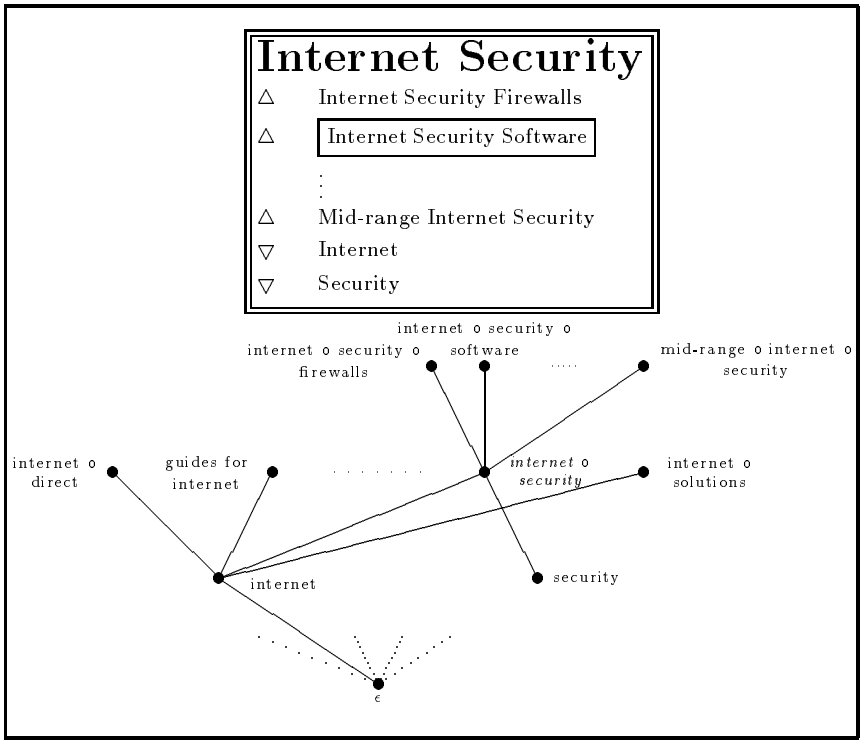

Figure 4: *After refining*

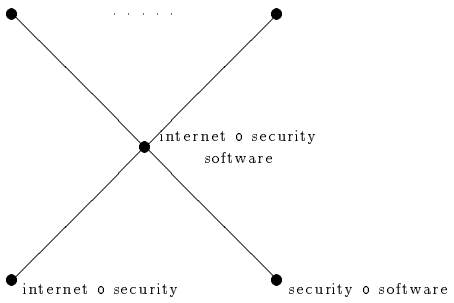

Figure 5: *Enlargement possibilities of* internet ∘ security ∘ software

it can be used as a query and the result presented to the user. Information discovery by a process of navigation through a hyperindex has become known as *query by navigation*[BW90]. Observe that in query by navigation refinements of a focus represent descriptions containing an *additional* term to the focus. In other words, if the focus is considered a query, the refinements represent possible expansions of it. By seeing the refinements the user is relieved of having to come up with their own terms to expand the query. This is particularly useful for the user who is not clear about their given information need. Enlargements, on the other hand, represent descriptions whereby a term has been *deleted* from the focus. In short, we believe that the refinements and enlargements represent potentially useful modifications of a particular focus (query).

The architecture of the hyperindex search engine is depicted in figure 6.

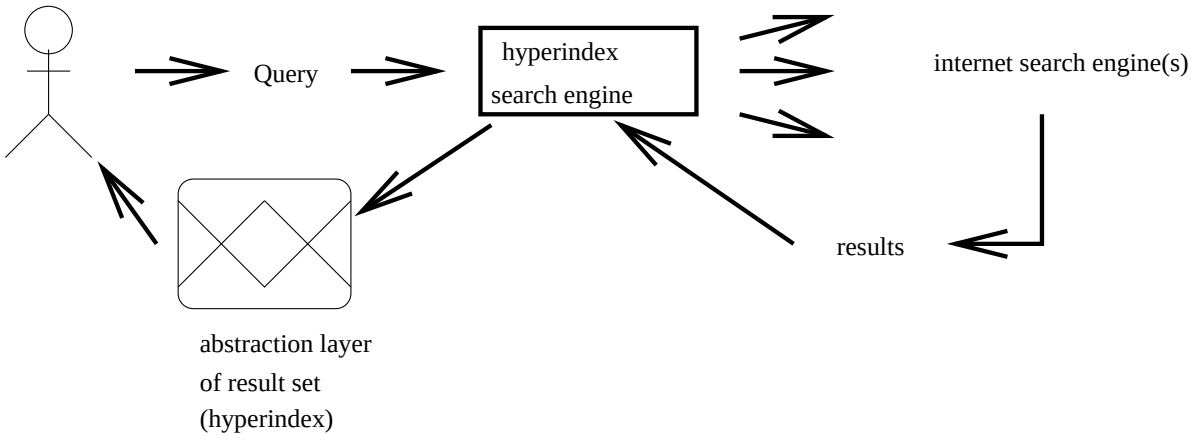

Figure 6: *Hyperindex Search Engine Architecture*

The interesting feature of the hyperindex search engine is the hyperindex. This will now be explored further.

3.1 Hyperindices

The index terms in the hyperindex have the form of so called *index expressions* [Bru93]. In contrast to keywords or term phrases index expressions have a structure (see figure 7). This

figure also shows that the relationships between terms are also modelled. These relationships are termed *connectors*, or *operators*, which are basically restricted to prepositions. Figure 8 shows some of the allowable connectors and the relationship types they denote.

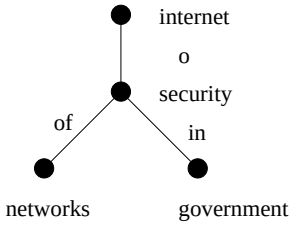

Figure 7: *Example Index Expression*

Connector	*Rel. Type*	*Examples*
of	possession action-object	castle of queen pollination of crops
by	action-agent	voting by students
in, on, *etc.*	position	trees in garden
to, on, for, in	directed association	attitudes to courses research on voting
with, ∘, and	association	Napoleon with army fruit ∘ trees
as	equivalence	humans as searchers

Figure 8: *Connectors and their associated Relationship Types*

From the structure of an index expression the so called power index expression can be derived by computing all index subexpressions. The power index expression results in a lattice. By way of illustration, the power index expression of the index expression depicted in figure 7 is shown in figure 9. (For convenience,"security" has been abbreviated to "sec" and "government" has been abbreviated to "govt"). The symbol ϵ signifies the empty index expression, which characterizes all information objects. Conceptually the hyperindex is a union of such lattices. This is the structure which supports query by navigation.

In practice, however, it is not necessary (or desirable) to generate the complete hyperindex. What is done is the following: When the query result from the search engine(s) is returned all titles are first extracted. Each title is passed through the index expression parser resulting in one or more index expressions. The resulting set S of index expressions from all the titles are the building blocks for the hyperindex. Let us assume that the query index expression q has n terms. The problem is to compute the set of refinements $refine(q)$ and the set of enlargements $enlarge(q)$ using the set S and query q. The refinements and enlargements are then presented to the user with the query q being the current focus of the search.

A refinement of q is the tree represented by q with an additional connector-term pair. In terms of figure 9, refinements of the index expression internet ∘ security has the refinements

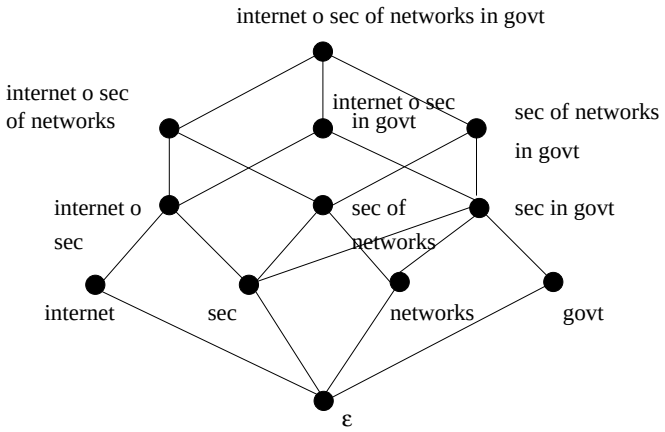

Figure 9: *Example Power Index Expression*

internet ∘ security of networks and internet ∘ security in government. Note that a refinement of an index expression of n terms has $n+1$ terms. The set $refine(q)$ can be computed by iterating through the set S and computing the subexpressions of $n+1$ terms of which q is a subexpression of.

Conversely, an enlargement, is the tree resulting by removing a connector-term pair from q. Consider figure 9. Enlargements of internet ∘ security are internet and security. Enlargements of an index expression of n terms is the set of subexpressions of $n-1$ terms.

On short, refinements and enlargements are derived from a tree structure. Essentially the tree structure represents terms which co-occur within the context of term relationship (the edge of the tree). If we view the refinements and enlargements as potential modifications of a query, then these modifications are being derived from an underlying linguistic structure rather than by term co-occurrence statistics as done in other approaches (see, for example, [QF93]).

3.2 The Derivation of Index Expressions from Title Descriptions

The input to the index expression parser is the titles of web pages. The first phase of the index expression parser is to remove stop words such as the and a are removed. For example,

Internet security in of networks in the government

results in

internet security of networks in government

While removing articles the parser also checks for so called connector irregularities. This occurs when there are two or more connectors between terms. As there may only be one such connector, the parser chooses the first. On the other hand, if there are no connectors between two successive terms, the null connector ∘ is inserted. Furthermore, connectors at the beginning of the title are removed. In the running example there are no connector irregularities:

internet ∘ security of networks in government

The problem remains to detect the underlying tree structure. This is achieved by employing a two level priority scheme in relation to the connectors. This priority scheme was a result of the observation that some connectors bind terms more strongly. The connectors deemed to bind most strongly are ones like ○,and,with and of. These connectors are deemed, perhaps counter intuitively, to have priority 0. Connectors such as in have been found to bind less strongly and receive priority 1 [Bru93].

The connector priorities are used to derive an underlying tree structure in the title currently being indexed. This structure is built up as the descriptor is scanned left to right. The heuristic used is that the tree is *deepened* if a high priority connector is detected, otherwise it is *broadened*, sometimes at the root, sometimes deeper in the tree.

Figure 10 shows the successive build up of the tree structure using the running example. Up to the point of parsing in the tree developed thus far has been deepened twice as both ○ and of are strongly binding connectors When in is parsed the tree is broadened as it is a lower priority connector.

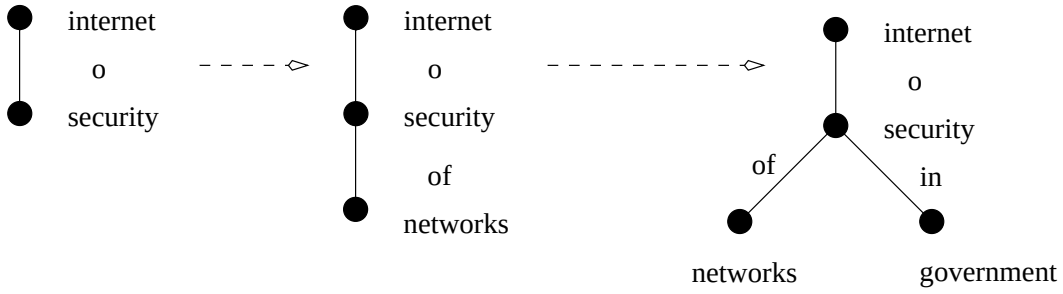

Figure 10: *Detecting structure*

3.3 Implementation of the Hyperindex Search Engine

There are currently two Hyperindex Search Engine prototypes[3]. It is important to note that the index expression parser was not originally developed for parsing titles on the internet [BBB91, Bru93]. Bosman et al report in their study that the parser had a ninety percent success rate in producing correct tree structures when parsing art-history titles [BBB91]. (These titles were in Dutch, German, Italian and French, as well as English). Similar success rates were attained when parsing titles from the CACM and Cranfield documents collections [Bru93].

Though no analysis has yet been performed with regard to the tree structures generated by parsing titles from web pages, our general impression is that the correct structures are still being produced more than seventy percent of the time. Most errors occur when title descriptions are not in the passive form, e.g. 'Clinton faces a hostile senate". An additional facet of the titles of web pages is their short length and lack of explicit connectors. The null connector ○ thus dominates many trees derived from web titles. This unfortunately degrades the structure of the underlying hyperindex as less refinements are produced. For this reason our HotOil hyperindex prototype sends the user's query, not only to internet search engines, but to the bibliographic database "Melvyl". The titles in bibliographic databases are of

[3]http://www.dstc.edu.au/cgi-bin/RDU/hib/hib and http://www.dstc.edu.au/cgi-bin/RDU/hotOIL/hotOIL

higher quality than on web pages resulting in a hyperindex with a richer structure. In fact, the hyperindex search engine leverages off the quality of the titles of bibliographic records offered by Melvyl to allow queries to be launched into the internet at large.

4 Conclusions and further research

In the first part of this paper the logs of the babyOIL search engine were analyzed to determine the proportions with which different types of query transformation occur. It was found that the primary transformation type was repetition of the previous query. Users also substitute, add and delete terms from a previous query and with lower frequency split compound terms, make changes to spelling, punctuation, and case and use derivative forms of words and abbreviations.

In the second part of the paper, the Hyperindex search engine was outlined. The Hyperindex search engine defines processes of term refinement and enlargement which add and delete terms, respectively, using a simple and widely applicable grammatically based formalism. The Hyperindex search engine has been successfully applied to a number of document collections including the World Wide Web and can use high quality domains such as Melvyl to index into low quality domains such as the World Wide Web, thus hopefully improving retrieval performance. The Hyperindex search engine is a manifestation of a two level hypermedia [BW90, Luc90, ACG91, AM92, GGP89]

Further research will centre around evaluating the hyperindex search engine for retrieval effectiveness, as well as investigating preference reasoning to allow the hyperindex to be dynamically structured according to user preferences for information. The basic idea is the following. As the user moves through the hyperindex they are essentially giving feedback as to what index expressions they find relevant. This feedback can be translated into *nonmonotonic consequence relations* which are a vehicle for representing preferences [BL97]. For example,

$$\mathsf{internet} \mid\!\sim_N \mathsf{security}$$

is a nonmonotonic consequence relation expressing that in the light of information need N, the preferred information objects about the internet deal with security.

The nonmonotonic consequence relations are fed into an inference system which allows index expressions to be derived that are consistent with the preferences expressed by the user via their particular navigation path through the hyperindex. The derivations resultant from the inference process could be fed into the user interface as additional refinements, or enlargements. In this way, the hyperindex search engine becomes a user sensitive search tool.

Acknowledgements

The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of Australia.

References

[ACG91] M. Agosti, R. Colotti, and G. Gradenigo. A two-level hypertext retrieval model for legal data. In A. Bookstein, Y. Chiaramella, G. Salton, and V.V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 316–325, Chicago, Illinois, October 1991. ACM Press.

[AM92] M. Agosti and P.G. Marchetti. User navigation in the IRS conceptual structure through a semantic association function. *The Computer Journal*, 35(3):194–199, 1992.

[BBB91] R. Bosman, R. Bouwman, and P.D. Bruza. The Effectiveness of Navigable Information Disclosure Systems. In G.A.M. Kempen, editor, *Proceedings of the Informatiewetenschap 1991 conference*, Nijmegen, The Netherlands, 1991.

[BL97] P.D. Bruza and B. van Linder. Preferential Models of Query by Navigation. In F. Crestani, M. Lalmas, and K. van Rijsbergen, editors, *Information Retrieval and Logic*. 1997. Forthcoming book chapter. Current version can be found at http://www.icis.qut.edu.au/~bruza/pubs.html.

[Bru90] P.D. Bruza. Hyperindices: A Novel Aid for Searching in Hypermedia. In A. Rizk, N. Streitz, and J. Andre, editors, *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 109–122, Cambridge, United Kingdom, 1990. Cambridge University Press.

[Bru93] P.D. Bruza. *Stratified Information Disclosure: A Synthesis between Information Retrieval and Hypermedia*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, 1993.

[BW90] P.D. Bruza and Th.P. van der Weide. Two Level Hypermedia - An Improved Architecture for Hypertext. In A.M. Tjoa and R. Wagner, editors, *Proceedings of the Data Base and Expert System Applications Conference (DEXA 90)*, pages 76–83, Vienna, Austria, 1990. Springer-Verlag.

[BW92] P.D. Bruza and Th.P. van der Weide. Stratified Hypermedia Structures for Information Disclosure. *The Computer Journal*, 35(3):208–220, 1992.

[GGP89] R. Godin, J. Gecsei, and C. Pichet. Design of a Browsing Interface for Information Retrieval. In N.J. Belkin and C.J. van Rijsbergen, editors, *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32–37, Cambridge, Massachusetts, June 1989. ACM Press.

[Luc90] D. Lucarella. A Model for Hypertext-Based Information Retrieval. In *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 81–94, Cambridge, United Kingdom, 1990. Cambridge University Press.

[QF93] Y. Qui and H.P. Frei. Concept Based Query Expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169, 1993.