Support Strategies for Interactive Thesaurus Navigation

Susan Jones and Micheline Hancock-Beaulieu, Centre for Interactive Systems Research, Department of Information Science, City University, London, England

Abstract: In principle, the "knowledge" encoded in a thesaurus can be exploited in many ways to help users clarify their information needs and enhance query performance, but attempts to automate this process via AI techniques face many practical difficulties. In the short term it may be more useful to improve support for direct interactive use of thesauri. We discuss some of the issues which have arisen when building an interface for thesaurus navigation and query enhancement, drawing on logs and user feedback from ongoing small-scale experiments.

1. Introduction

A thesaurus represents a collection of organised knowledge, often based on an abstract classification scheme which provides a "map" of some subject domain. Traditionally it has been used as a source of preferred terms for document indexers and, for intermediaries and expert end-users, as an aid to the precise definition of an information need in a controlled search language.

The rapid growth in the number and size of available text databases over the last few years has affected traditional views as to the role of the thesaurus. On the one hand it is argued (Oppenheim 1993) that the time and effort involved in thesaurus construction and controlled-language indexing is no longer economically justifiable, now that in practice most users of public on-line services make free-text searches on full-text databases. Conversely (Pollitt 1990), thesaurus structures are sometimes taken as the framework for new "intelligent" menu-based front-ends to IR systems. Much research has gone into the enrichment of traditional thesauri (e.g. by assignment of "weights" to inter-term relationships) so as to support semi-automatic query expansion procedures (Chen 1991).

Meanwhile, organisations increasingly feel the need to develop specialised thesauri to control and access their own internal resources. They provide a market for software suppliers who now routinely include a thesaurus component in their library products, often apparently designed with scant regard for the kind of data actually present in standard thesauri, or the conceptual problems which they pose for end-users.

Following is an account of some work carried out within the City Interactive Knowledge Structure (CILKS) project, funded by the Joint Information Systems Committee of the HEFC under the Knowledge-Based Systems Research Initiative. It traces the development of a prototype designed to support interactive thesaurus navigation, and draws out some of the lessons learnt from small-scale evaluation experiments with users. The task of finding useful terms in a thesaurus is an adjunct to the primary task of finding useful documents in a bibliographic database, and many of the same techniques and problems are applicable.

2. Prototype development

The overall project aims were to exploit thesaurus knowledge in the construction of an IKBS to enhance probabilistic information retrieval. It was decided that a typical thesaurus should be loaded into a relational database, enabling researchers to explore and analyse its characteristics thoroughly before proceeding further. To that end a data model was designed (Jones 1993), and a database set up using the Oracle DBMS software, and containing the INSPEC thesaurus. Early investigations used standard SQL queries to generate reports and statistics. A prototype interface based on Oracle SQL*Forms was then created to allow more selective explorations of the data, and, later, proper user-driven query expansion as a preliminary to document retrieval. Over the last two years this has been refined in response to feedback from small-scale supervised experiments with users, so that is is now sufficiently robust to act as a tool for investigating thesaurus strategies.

It should be stressed that interface design has never been a primary objective of this research. What was needed was a piece of software which would allow users to consult the thesaurus in order to produce an expanded query for submission to Okapi - a "best-match" retrieval system based on term weighting and relevance feedback (Hancock-Beaulieu & Walker 1992). Behind the scenes, all details of choices made and terms selected would be logged in the database for later analysis. Thus it was hoped to identify patterns of behaviour in thesaurus navigation leading to successful retrieval which could later be built into automatic query expansion procedures.

So far such patterns have proved elusive (Jones et al.1993), but useful facts have emerged about the functions which need to be provided by the thesaurus-handling component of an interactive IR system. In addition to the automatic logs, information is collected by questioning users before during and after their sessions with the thesaurus. As the interface improves, comments about the practical difficulties of interaction give way to more interesting insights into the nature of the task, and what exactly is happening when a query is expanded in this way.

The software as it stands is still by no means ideal; many users have expressed a preference for a GUI interface rather than one based on forms and function keys, and it is intended to move in that direction in future. However the prototype has enabled us to learn some important lessons before that work is undertaken, although it is worth saying in passing that, for an application requiring tight control over complex processing logic, the development effort with a 4GL product like Oracle SQL*Forms is not much less than that required for conventional 3GL programming.

2.1 Starting points: query-term matching

The crucial factor for success in thesaurus-based query expansion is finding a good match between the original query and one or more controlled language terms. Since most queries, and about 75% of terms, contain more than one word, a "best-match" algorithm is applied (after stemming), using weights based on thesaurus word frequency. Possible matches are displayed to the user for selection in weight order, so that the most likely candidates appear at the top of the list. This is the part of the interaction which uses most processor time, as in principle every one of the 7000-odd words occurring in a thesaurus term must be compared with every word in the query. In practice the database is very heavily indexed, and in more
recent versions of the prototype the matching algorithm is coded in C rather than SQL to give faster response.

The first version of the system showed all potential matches between query and thesaurus terms, even those based on a single extremely frequent word like "systems". This made the interaction very laborious - many users complained about the number of irrelevant terms seen on the first screen; some ignored all but the top few while others used their time unfruitfully scrolling through the hundred or more items on the full list. With each successive version of the software the cut-off point has been raised, so that currently there are never more than thirty terms displayed after the first match.

Inevitably there is some trade-off here. During an early experiment one user entered the query: "help desk systems". Neither "help" nor "desk" appeared as matching words, but by patient searching the user picked up "system documentation", which led her to "technical support services", "information services", and "information centres" - all relevant to her need. In the current version this particular match might not occur, but it appears that in most cases the first match does identify at least one useful starting point for navigation.

2.1.1 Alternative access via class-codes

Given the importance of the query-term match, a small separate experiment was conducted (Fong 1993) to test an alternative "way in" via descriptors associated with the Inspec class-codes. Unlike most standard thesauri, Inspec has a class-code hierarchy which is separate from the term hierarchy; each class-code has its own verbal descriptor, and there is a many-to-many relationship between thesaurus terms and class-codes. By matching in the usual way between queries and class-code descriptors, it was possible to find one or more potentially relevant codes, and hence the terms associated with those codes.

The experiment certainly proved the viability of the method, and its potential as a "fall-back" when basic query term matching fails. In fact, however, the descriptor vocabulary is much smaller than the term vocabulary (2466 words as against 6987, of which 598 are common to both, and with a much higher proportion of stop-words). So if a query does not match any thesaurus term, it is tolerably unlikely to match a class descriptor. Moreover many class descriptors at the top of the hierarchy are very general, so cannot match well with specific queries.

It was noted also that the term-code link effectively defines another, very loose, "synonym" relationship between terms, and thus an additional navigation route through the thesaurus. Some comparative results were produced, showing that the effect of using this route is to increase recall rather than precision: a low proportion of terms found is likely to be selected, although the method does reduce the lengths of the search paths needed to find some relevant terms. In practice most standard thesauri specify a one-to-one relationship between class-codes and terms, so the method is not readily generalisable outside the Inspec context, and has not been pursued further in the main project.
2.2 Interactive thesaurus navigation

Some of the earliest design decisions were about how to present the interface as a convenient tool for building up a collection of terms for query expansion, rather than simply viewing the thesaurus. It was decided that users should not need to choose which relationships to follow from a given term, nor be limited in the "distance" they might travel, (although in practice they tend not to make more than four or five moves from their original starting-point).

Following an initial selection of terms matching the query, the user may switch repeatedly between two main states. One screen, headed "These are the terms which you have chosen so far", displays all the currently selected terms. From this screen it is possible to: save terms and exit, delete terms (or undo previous deletions), or select a single term for expansion. In the last case, all paths from the selected term are followed simultaneously. The other screen displays the result of expansion by showing all the broader, narrower, and related terms, any number of which may be selected and added to the current collection before returning to the previous screen.

The database contains some information which would not be available to someone consulting a printed version of the thesaurus. On the two screens described above, each term is shown on a separate line along with the following data:

- **Status**: classifies the term as being either a lead-in, or, if a preferred term, as having a top, mid, or bottom position in a hierarchy.
- **Num-rels**: the number of immediately related terms.
- **Num-docs**: the number of postings for the term in the associated document database.

Some users have indicated that the above information does help their decision-making, particularly if they are consciously looking for more general or more specific terms. However it sometimes has to be explained that term specificity varies according to a particular thesaurus’s coverage - one user was disconcerted to find that in Inspec "psychology" was a bottom term with nothing more specific below it!

2.2.1 Handling lead-ins and duplication

One thing which has persistently caused confusion to users, (and logistic problems to the software developer), is that it is so easy to get to the same term by several different routes. The Inspec thesaurus contains many lead-in terms (7023 as against 6454 preferred terms), and, contrary to the ISO standard, allows many-to-many relationships between lead-in and preferred terms. Moreover a lead-in and its associated preferred term often have words in common, so that both are likely to be identified by a partial match with the same query.

When a lead-in term is chosen the system brings up a subsidiary screen showing all its preferred terms - in theory the user may pick any which appeal to him and they should replace the lead-in in the main displayed list. However, as already suggested, some of those preferred terms may well be on the main list, and indeed selected, already. Much confusion arose in early experiments because users could not remember what they had selected, chose the same term twice or left it out because they thought they had it, and so on. Similar problems occur later after term expansion, when by definition at least one and often more terms appear which have already been seen and selected.
Over the development period, it has been found necessary to modify the interface several times to handle these problems. Actually identifying all cases of potential duplication requires some complex background processing. The original approach was merely to show that a term was selectable or had been selected elsewhere, by marking it with an asterisk. The stronger policy now adopted is to disallow selection of marked terms altogether. This sometimes disconcerts users, but it has the merit of consistency and does prevent the duplication which occurred in the past. A more flexible and informative approach would of course be easier to implement in a multi-window GUI environment.

2.2.2 Handling faceted queries

The most recent innovation deals with queries in which two or more distinct topics intersect. (See Jones et al 1993 for a preliminary discussion of the principles involved.) After entering a query, the user is invited to rephrase it as two or three separate facets - if this is accepted each facet is matched independently with possible thesaurus terms, and facet groupings are maintained by the system during the subsequent navigation and term collection. The display to the user, and the background logs, have been modified to record a complete derivation history; some example reports based on this data are shown in the following section. On exit from the thesaurus system, a script converts the set of terms into a structured query which will enable the Okapi term-weighted retrieval system to take the facet groupings into account.

This facility has been made available only recently, and little feedback about it is yet available. At the time of writing it is possible to report that five out of the first ten users took up the invitation to partition their queries, and that three of these finally generated an expanded query comprising more than one facet group. The concept did not apparently give rise to confusion, at least under supervised conditions. Its effect on query enhancement will be evaluated when the current phase of data collection has been completed.

2.3 Some expanded queries

The following tables illustrate three typical navigation "histories" for users of the most recent version of the software. Each term is shown below and one place to the right of the term from which it was derived. It is preceded by a symbolic representation of the "path" of relationships followed to reach it. The symbols < > - stand for BT, NT and RT respectively. So, for example the path "--<" indicates that the term in question was reached by following two associative links and then an upward hierarchical link.

The first example shows how a single-word query ("hypertext") was expanded to twenty terms, following navigation up to six moves away from the starting point. The user classified his original query as "moderately broad" - not surprisingly his comment on the final version was that it was "probably broader because of the new terms",and that "it might cause a change of direction" in his search.

The second example perhaps shows what ought not to happen during thesaurus navigation: a journey via five associated links from the lead-in term "teleworking" to "earthquakes"! At the time the user was able to offer a rationale for her choices here (she was seeking possible reasons why people might choose to work from home), but it is unlikely that her original query has been usefully expanded.
The final example shows faceting in action, with two large independent groups formed from the original starting points: "expert systems" and "education".

Example 1

hypermedia
- electronic publishing
-- multimedia systems
-- computer applications
-- information science
-- document delivery
-- information analysis
-- information dissemination
-- information needs
-- information retrieval
-- information services
-- information use
-- information industry
-- information systems
-- public information systems
-- information retrieval systems
-- online front-ends
-- publishing
-- microforms
-- telecommunication services

Example 2

home working
- social aspects of automation
-- security of data
-- computer installation
-- disasters
-- geophysical catastrophes

Example 3

expert systems
< knowledge based systems
< intelligent tutoring systems
-- artificial intelligence
-- learning
-> Hebbian learning
-> backpropagation
-> learning by example
-> unsupervised learning
-- learning systems
-- natural languages
-- explanation
-- user modelling

education
- computer aided instruction
-> computer based training
-- teacher training
-- training
-- professional aspects
computer science education
educational computing
educational aids
educational courses
psychology

3. User perceptions

Evaluation of the prototype (and of various formulae for query expansion) is proceeding via a series of small-scale trials with typical academic users of the Okapi retrieval system. Two experiments have been run; the third is in progress. Detailed quantitative results from the first two are given in (Jones et al 1994) - they can be summarised by saying that a) no strong patterns of behaviour emerged as the basis for possible automatic query expansion procedures, and b) the effect of using the thesaurus was to bring new relevant documents to light, but sometimes to the detriment of query precision at higher document cut-off levels. The current discussion focuses on information gathered about users' perceptions of the thesaurus and its effect on query formulation:

Ideally, the thesaurus navigation task should be interesting and enjoyable, and in fact a few users clearly find "exploring the database, finding new terms" a pleasant experience. Unfortunately a larger number find it frustrating - whether because of poor response times,
inadequate coverage of the subject area ("a good idea in principle but I could not find the terms I wanted"), or problems of orientation ("it sends one off at a tangent", "no related text to give a context to the terms", "I would prefer to see a structural / graphic display").

One point of interest is whether users with an information need feel that consulting the thesaurus and making relevance judgements on retrieved document actually improves their subject knowledge. Accordingly they are asked "How well do you know this topic now?" three times: at the start and end of the thesaurus session and again after scanning the abstracts in the retrieved document list. Most users register "no change"; a minority report improvement (either after thesaurus navigation or after after scanning documents); a very small number indicate that they know the topic less well by the end, suggesting perhaps that their earlier confidence was misplaced.

Users are also asked to rate their original query as "broad", "fairly broad", "narrow", etc., and questioned about their satisfaction - with their own query and with new terms gathered from the thesaurus. They tend to rate newly-found terms less well than their original query, although at this point are probably not making a direct comparison. An interesting range of opinions emerges when direct comparisons are sought, particularly with regard to query breadth. (See the table of example responses below.) Many of those who think their original query was "broad" consider that thesaurus search has made it more specific, even though more terms have been added. One user states explicitly that both broadening and narrowing have occurred!

<table>
<thead>
<tr>
<th>Original query</th>
<th>Comment on new query</th>
</tr>
</thead>
<tbody>
<tr>
<td>very broad</td>
<td>improved - contains new terms not previously thought of</td>
</tr>
<tr>
<td>very broad</td>
<td>expanded to a useful extent</td>
</tr>
<tr>
<td>very broad</td>
<td>more specific</td>
</tr>
<tr>
<td>very broad</td>
<td>more refined</td>
</tr>
<tr>
<td>very broad</td>
<td>more clearly defined</td>
</tr>
<tr>
<td>very broad</td>
<td>comprises only one &quot;strand&quot; of original topic</td>
</tr>
<tr>
<td>broad</td>
<td>very narrow part of original</td>
</tr>
<tr>
<td>broad</td>
<td>narrowed down - &quot;correcter&quot; than original</td>
</tr>
<tr>
<td>broad</td>
<td>much narrower search using Inspec's &quot;language&quot;</td>
</tr>
<tr>
<td>fairly broad</td>
<td>narrower AND broader - more specific terms</td>
</tr>
<tr>
<td>fairly broad</td>
<td>but a larger number of them</td>
</tr>
<tr>
<td>fairly broad</td>
<td>still too general; some relevant terms</td>
</tr>
<tr>
<td>fairly broad</td>
<td>not as detailed</td>
</tr>
<tr>
<td>narrow</td>
<td>less specific than the original - led away from what I was after</td>
</tr>
<tr>
<td>very narrow</td>
<td>similar, did not find many new terms</td>
</tr>
</tbody>
</table>

This is not as paradoxical as it seems. For boolean queries, the addition of new terms implies also selection of AND / OR operators which do explicitly broaden and narrow. In a weighted retrieval context, it is hoped that new terms will bring new documents into consideration, while promoting the most relevant documents to the top of the ranked list. That has in fact been the rationale behind the whole CILKS project. Users are never directed to broaden or narrow their queries, but to select any terms which might represent some aspect of their information need. Perhaps the most frequently-repeated comment is that the sessions reveal "new terms not previously thought of" - the essential raison d'etre of a thesaurus.
4. Future developments

Although evidence gathered by questioning users in this way is quite impressionistic, it has helped to point the way forward. As already noted, a feeling of confinement at being able to see only a small part of the search space is quite common; many users would like the thesaurus to act as a high-level map of a larger domain. For that purpose, a hierarchical interface of the type described by Pollitt may be beneficial, providing a possible alternative to query-term matching via individual words as a way in to the relevant part of the thesaurus.

So far, because the central focus of our work has been the attempt to improve the performance of the initial Okapi best-match search, all query expansion has taken place before document retrieval. The trial conditions enable us to make easily-controlled comparisons between original and expanded queries, but are rather artificial from the user’s perspective - some users are understandably reluctant to judge the value of new terms without seeing any real documents. Current development work to close the gap between the thesaurus prototype and interactive Okapi will shortly allow a more natural movement in each direction between the two databases, and the possibility of query re-formulation after retrieval.

Our eventual goal is an integrated system accessed via an extended version of the current Okapi GUI (Fieldhouse et al 1994) - in this context thesaurus search will take its place as one amongst a number of possible techniques for query enhancement. For perhaps the most conclusive result from our experiments so far is that the thesaurus on its own, however good its coverage, will not always be the best mechanism for matching queries to documents. On average, simply substituting the best set of thesaurus terms for a free-text query does not improve performance - too much information is lost. A more fruitful approach has been to generate hybrid queries containing both free-text and controlled language terms; and even here it appears that measures which enhance one set of queries may have an adverse effect on others. That, however, reflects the generally elusive nature of the information retrieval task.

References


