A Visual Information-Retrieval Navigator

Shalini Sewraz and Stefan M Rüger
Department of Computing, Imperial College
London SW7 2BZ, England

Abstract

We present a visualisation front-end that aids navigation through the set of documents returned by a search engine (hit documents). Our method is based on a clustering of the hit-document set. We have overcome the curse of dimensionality by representing each hit document with a small vector that is a histogram of related terms such as "software", "UNIX", "IBM", "users" for the query "computer". We compute these related terms dynamically from the subset of hit documents. The obtained clusters based on this representation have proven to be meaningful. We make use of the clustering to visually group the documents returned from the search and label the groups with their respective related words. The navigator can browse cluster information as well as drill up or down in one or more clusters and refining the search using one or more of the suggested related keywords. Our prototype is designed for novice, typical and expert users so they can take advantage of whichever search method they are comfortable with.

Introduction

Although the computer is a powerful tool for searching, most conventional search engines are plagued by low precision returning thousands of hit documents as their output. A common problem with this is that users have to wade through much non-relevant material before finding relevant documents.

Search results could be improved by *query refinement* which means augmenting the query with additional search terms after the initial search. This is in fact another interesting and well-known recent advance of AltaVista, called Live Topics. This feature is useful in narrowing down a search; however, the way it is presented in AltaVista the search result is still shown as a long list of pages to browse through. In our opinion, the strategy should be to shift the user's mental load from these slower thought-intensive processes such as reading to faster perceptual processes such as pattern recognition in a visual display. The page metaphor, though simple, is too restrictive: with large volumes of data displayed on multiple pages we find ourselves searching all over again!

Furthermore, in conventional search engines including AltaVista's Live Topics the documents are ultimately ranked with the aim to order them according to relevance to the user. This appears to be overly ambitious as even advanced ranking algorithms cannot know whether the user prefers documents about "hardware" or "software" when the query simply was "computer". We suggest displaying clusters of documents. Cluster Analysis itself is a technique which assigns items to automatically created groups based on a calculation of the degree of association between items and groups. In information retrieval cluster analysis has been used to create groups of documents, based on the terms they contain, with the aim of improving the efficiency and effectiveness of retrieval [13]. Indeed, the Cluster Hypothesis of Information Retrieval states that "closely associated documents tend to be relevant to the same request" [18] implying that if one document is relevant to a query then it is rational to include other similar documents. Such document clustering would thus be useful for separating relevant and non-relevant documents.

Post-retrieval document clustering has been well studied in the recent years, see eg [7, 1, 12, 21], and many methods of information visualisation have been described, see eg [10, 5, 17, 8, 2, 3]. We have contributed and evaluated a new feature reduction method for post-retrieval clustering and a corresponding

visual navigator. Our process of computing related words for a particular query, the representation of documents as small histogram vectors of related words and the corresponding clustering of documents has been described elsewhere [22, 14] and is summarised in subsections 2.1 to 2.3. Subsequently we will concentrate on the design and the implementation of our information-retrieval navigator.

1 Functional Requirements

The user has to execute at least two types of activities: The first is related to the *navigation task* during the search by moving and exploring information. The second category of activities is related to the *information task* that has to be executed during the search. This includes clustering of documents, which will hopefully place similarly relevant documents in a single cluster providing an overview of the retrieved document set and helping the user locate interesting documents more easily. It also includes judging if it makes sense to continue the search in a particular direction. Thus some method to inspect information items in more detail is also required.

Moreover, as users query and browse, more is learnt about the problem and potential solutions, thereby causing a refinement of the conceptualisation of the problem. Thus the problem of finding relevant information evolves and is refined through the process of seeing results of intermediate 'queries', with browsing itself helping to facilitate the iterative and ill-defined nature of information seeking [9].

Finally, guidelines for the design of such systems must not only address issues of look and feel but also of effective interaction relevance. For instance feedback and query reformulation explicitly address the ill-defined nature of information seeking by allowing users to learn from the repository and iteratively refine the information need.

Thus the system should also support a number of interaction styles such as browsing and querying to accommodate the different kinds of search strategies users may need to use. Hence, in summary, the functional specifications are:

- run-time interactive (dynamic map)
- enable browsing
- easy and simple navigation
- minimise reading by providing a visual representation of data objects
- avoid high dimensional space
- provide a common look and feel for all textual and graphical representations
- refined search
- provide two levels of explanations - a quick summary and further details
- drill down to next level versus whole picture
- simple - no special knowledge required

2 System Specifications

We implemented a system as shown in Figure 1.

2.1 Query Results and Related Words

In the vector model documents are represented as histogram vectors of their words. A document collection can contain millions of different words. Even after removing the function words of the language ("the", "with" etc) and even if only nouns are kept that appear in not too many documents (otherwise they are not informative) and in not too few documents (otherwise they seem too specialised), the vocabulary of a collection can easily be in the 100,000s; we call these words the *potentially interesting words*.

Even a subset H of documents that is returned by a search engine upon an initial query can easily contain 10,000s of potentially interesting words. A document representation in this subset H would be of the same

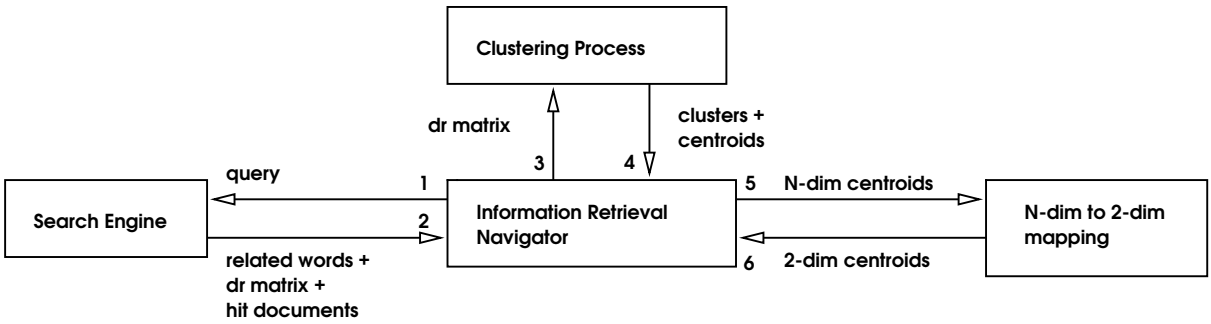

Figure 1: Flow diagram of the implemented system

dimensionality. The problem with this kind of vector is that any two randomly picked vectors in a high-dimensional hypercube tend to have a constant distance from each other, no matter what the measure is! As an example, let $x, y \in [0,1]^n$ be drawn independently from a uniform distribution. The expectation value of their sum-norm distance is $n/3$ with a variance of $n/18$. For $n = 1,800$ (corresponding to a joint vocabulary of just 1,800 words for a word histogram representation) this means a typical distance of $|x-y|_1 = 600 \pm 10$. With increasing n the ratio between standard deviation and vector size gets ever smaller, as it scales with $1/\sqrt{n}$. Although word histogram document representations are by no means random vectors, each additional dimension tends to not only spread the size of a cluster but also dilute the distance of two previously well-separated clusters. Hence, it seems prohibitive to involve all semantic features (eg the words) of a document collection for document clustering.

Document clustering has attracted much interest in the recent decades, eg [15, 6, 19, 13], and much is known about the importance of feature reduction in general, eg [11] and clustering in particular [18], but little has been done so far to facilitate feature reduction for document clustering of query results.

We suggest ranking the importance of each such word j with a weight

$$w_j = \frac{h_j}{d_j} \cdot h_j \log(|H|/h_j),$$

where h_j is the number of documents in H containing the word j, and d_j is the number of documents in the whole document collection D containing j. The second factor prefers medium matched-document frequency h_j, while the first factor prefers words that specifically occur in the matched documents. The highest-ranked words are meant to be related to the query. Indeed, we have "software", "IBM", "UNIX" etc as the top-ranked words when querying for "computer". This seems to be a powerful approach to restrict the features of the matched documents to the top k ranked words, which we will call the *related words*. One important aspect is that the features are computed at query time. Hence, when the above query is refined to "computer hardware", a completely new set of features would emerge automatically.

2.2 Document Representation and Feature Reduction

For each matched document i we create a k-dimensional vector v_i, where the j-th component v_{ij} is a function of the number of occurrences t_{ij} of the j-th ranked related word in the document i:

$$v_{ij} = \log_2(1 + t_{ij}) \cdot \log(|D|/d_j)$$

This is a variation of the tf-idf weight that stresses the term frequency t_{ij} less. We project the vector v_i onto the k-dimensional unit sphere obtaining a normalised vector u_i that represents the document i. We deem the Euclidean distance between u_a and u_b a sensible *semantic distance* between two documents a and b in the document subset H returned by a query with respect to the complete document collection D.

u may be viewed as a document set representation matrix (called dr matrix in Fig 1) where the row vector u_i is a k-dimensional representation of document i and u_{ij} is viewed as the importance of related word j for document i. In particular, $u_{ij} = 0$ if and only if i does not contain j. The number of features k can be controlled by us and our experiments have shown that $k \approx 10$ yields superior clustering results [22, 14]. Note that even if only the top ten related words are used for the clustering and document representation, we might still display more related words on the screen to assist the user in his/her search.

2.3 Clustering

Our system will then use a clustering process, which intakes the document set representation matrix and outputs clusters of documents. Each cluster contains a certain number of document vectors and is represented by their arithmetic mean, the so-called centroid vector. The distance between two centroids represents the similarity of the corresponding two clusters. Our clustering algorithm consists of two phases: in the first phase, hierarchical clustering with complete linkage operates on the best-ranked, say 100-150, documents. This can be done in a fraction of one second CPU time. Hierarchical clustering has the advantage that one can either fix the number of clusters one wants or let the number of clusters be determined by demanding a certain minimum similarity within a cluster. Either way, once clusters within the top-ranked documents are identified, their centroids can be computed and used as a seed for iterative clustering of the remaining documents. This algorithm consumes an amount of time linear in the number of documents and in the number of clusters. 1,000s of documents can thus be clustered in less than a second.

2.4 Two-Dimensional Representation

Finally, Sammon mapping [16] is used to convert these high-dimensional centroid vectors into two dimensions, while trying to preserve the distance among the clusters. These two-dimensional cluster vectors will ultimately be mapped onto the interface, thereby providing a visual landscape for navigation. Clustering cannot be performed in advance on the collection as a whole, as the features that encode a document are the related words which depend on the query (indeed, clustering should not be performed in advance as the hit documents returned by a query should ultimately determine how these documents are best projected).

3 Interface Organisation

As discussed in Section 1, the design should cater for novice users to begin working with little training but should still provide expert users with powerful features. We would thus like to cater for all types of users and search strategies. Firstly, there is the conventional or novice user making use of a simple search and result list. We use a standard-search-engine interface-type based on a page metaphor as in AltaVista, Yahoo, Excite etc for this purpose.

Then there is the user wishing to refine the search by means of suggested related words, especially in the case of a large list of documents retrieved. This case is dealt with an additional panel displaying related words and allowing the user to check terms to be included in the query (see Figure 3).

Finally, there is the user who wishes to explore, discern patterns among documents, browse and finally retrieve the relevant documents. This panel is a visual navigator of clusters with the aims of browsing, refining the search and retrieving (see Figure 4). This panel is divided into 3 parts. The right upper panel displays the clusters of documents in a two-dimensional space while the left upper panel displays the related words for each cluster and enables the refined search. Finally the bottom panel shows the resulting list of documents. This is further expanded on in the sections below.

4 Cluster Visualisation

Upon an initial query in the first panel the hit documents are grouped into clusters represented by centroid vectors which are projected onto this panel. It is to be noted that bringing higher dimensionality down to lower dimensionality for displaying is a trade-off between precision and cost. Lower dimensionality means somewhat rougher representations of document relationships but cheaper access and manipulation, the latter of which is more important here. The attributes and functionality accompanying these clusters are described below.

- Shape and quantity

  Each cluster is represented by a circle on the screen. The two-dimensional vectors obtained by the Sammon process were scaled down to fit the screen. To prevent cluttering of the screen it was decided to set the maximum number of clusters at any level to be a fixed number. Taking into account the screen size and the trade-off between having more specific clusters and cluttering the screen, it was decided to set this number to six. This is by no means a magic number and in future versions this number may well be set by the user within this panel.

- Distance

  The distance between any two circles in the panel represents the similarity of their respective clusters: the nearer the clusters, the more likely the documents contained therein will be of similar context thereby enabling the user to rapidly find all similar documents.

- Size

  It was also thought that an indication of the number of documents per cluster, in other words the size of the cluster, would be useful information for the user. For instance, if there are too many documents in that particular cluster, instead of wanting to see all the documents, the user could refine the search within this cluster thereby minimising the frustration and time taken in finding the document. Hence, the size of each cluster is represented by the size of the circle, with a maximum size being fixed for the biggest cluster and the rest being scaled accordingly. This is to prevent exceptionally large or tiny circle sizes displayed on the screen.

- Colour

  For the time being it was decided to use just one colour, with darker shades being used to identify clusters of greater volume and lighter shades being used to identify clusters with a smaller number of documents. As this is a redundant characterisation, custom-designed versions of this navigator may define another meaning for colour.

- Tooltip box

  It is not uncommon for a session to move across the spectrum from browsing to searching. Indeed, each new piece of information users encounter gives them new ideas and directions to follow and consequently a conception of the query [4]. For either purpose, it was deemed useful to provide a *Tooltip box* which contains additional information about each cluster (such as top-five related words of this particular cluster, number of documents) and which appears when the mouse cursor is positioned over a cluster (see Figure 2). Also, operations such as *Select*, *Drill up* and *Drill down* can be executed for this particular cluster or selection of clusters.

- Keyword refinements

  As with the second search panel, keyword refinements are possible within clusters or across a selection of clusters.

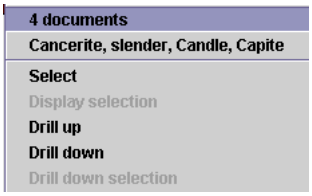

Figure 2: Capture of the tooltip box

- Display documents or a selection of documents

  When browsing through the clusters and identifying an interesting cluster, the user will probably want to see the document titles and descriptions contained in that cluster. As this should be a quick action, simply clicking on a cluster will display in the bottom panel a list of the document titles, descriptions and URLs. Similarly, a user might want to see at the same time the list of document titles and descriptions for more than one cluster. As this will be done after some thought and selection of relevant clusters, this option is in the tooltip box, and upon clicking on *Display Selection* this information will be displayed in the bottom panel.

Table 1 summarises the main features of each panel.

Panel	Features	Purpose
Simple	• Text based search • only 15 documents per view	• simple for the novice user • neatness; less tedious to read
Keyword	• suggests related words • allows refinement	• to inform about types of documents • to narrow down the search space
Cluster-based	• cluster display • fixed quantity of clusters • size and colour • distance • tooltip box • display documents • flags • drill down • drill up • general refined search • global/specific search	• visualisation for navigation • avoid cluttering • to show volume of data • similarity between clusters • provide summary information • provide detailed information • enable selection of interesting clusters • sifting interesting information • enables backtrack to higher level • narrow down the search • provide different types of search

Table 1: Features of the three panels

5 Evaluation

We evaluated the prototype system at two levels. First, we studied and quantified the effectiveness of the clustering method, ie, the ability to separate relevant documents from irrelevant documents. Then we looked at the effectiveness of the visualisation interface and the subjective judgement whether users find it useful.

For level one, we followed the ideas in [22] and performed experiments to assess the quality of the clustering process based on human-expert data. We used the 1997-1998 collection of the TREC data [20] with 528,155 documents, mainly US-American newspaper articles, 100 queries and corresponding relevance

assessments. The results showed a compelling evidence for the validity of the Clustering Hypothesis for post-retrieval document sets [14].

In order to evaluate the system at the second level, both heuristic and empirical approaches were followed. The heuristic evaluation was based on the user interface guidelines and functional requirements and showed good results. The empirical approach was based on user feedback by means of a questionnaire. This consisted of 3 parts. Firstly, there are a few questions concerning the users' computer abilities and familiarities with general search engines. In the second part, the user is asked to conduct a search using a conventional search engine and to answer a series of questions concerning the efficiency/effectiveness of the search, time spent, satisfaction and any suggestions. Finally, the user is asked to conduct the query using the Visual Information Retrieval Navigator and to answer a similar set of questions, including whether it assists in narrowing down the search. The questionnaire was distributed to a number of users of varying computing abilities and their feedback analysed.

The conventional search engines were found to be familiar and easy to use. However, it was agreed that this often led to unsatisfactory results and to having to read through long lists with a lot of irrelevant material. Moreover, the optional boolean search feature (AND/OR or +/− notation) was not found to be very popular. Concerning our new system, generally the familiar query input box and push button was appreciated as it gave the user confidence, especially novices. These users often preferred the simple page metaphor, though they found the clustering panel to be visually appealing; also they felt it prompted browsing. The latter opinion was held by most users. The clustering panel turned out to be the most appreciated panel, owing to the drill down/up feature, though the keyword-based search was found to be useful in narrowing the search as well. It was also found, mostly by expert/experienced users, that while browsing the clusters, the formulation of the initial query changed as different avenues were explored. Some features such as the combined use of keyword-within-clusters, was more appreciated after some familiarity with the system. Suggestions included displaying the context of each cluster in the tooltip box, labelling the cluster with its main keywords and making more use of the mouse buttons for the more frequent actions such as drill up/down.

Based on the user feedback, it is seen that the goal of catering for all levels of expertise was achieved by enabling new users to ignore the advanced features, while allowing more experienced users to explore these avenues. Expert users were quick to realise not only the potential of the clustering for a better sifting of documents but also the benefits of the visualisation in a more effective and user-friendly presentation of the information.

6 Conclusions

We have built an interactive visual information-retrieval navigator that displays hit documents grouped into clusters. This helps users narrow down their search by browsing the clusters first and then drilling down the relevant cluster. Alternatively, the search can be refined by selecting related words within one or more clusters. In order to cater for different categories of users the information-retrieval navigator has two other text-based panels in addition to above visual cluster navigation. These are similar to ordinary search engines with some added functionalities such as related words refinement. We are convinced that this sort of interface helps the human user to quickly narrow down their search based on a lazy and coarse initial query by analysing and categorising the available documents.

References

[1] R B Allen, P Obry, and M Littman. An interface for navigating clustered document sets returned by queries. In *Proc of the ACM Conf on Organizational Computing Systems*, pages 166–171, 1993.

[2] M Ankerst, D Keim, and H Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *IEEE Visualization '96*, 1996.

[3] J Assa, D Cohen-Or, and T Milo. Displaying data in multidimensional relevance space with 2d visualization maps. In *IEEE Visualization '97*, 1997.

[4] M Baldonado and T Winograd. Sensemaker: An information-exploration interface supporting the contextual evolution of a user's interests. *CHI Electronic Proc*, 1997.

[5] M Chalmers and P Chitson. Bead: Explorations in information visualisation. In *Proc of the 15th Intl ACM SIGIR Conf*, pages 330–337, 1992.

[6] W B Croft. *Organizing and searching large files of documents*. PhD thesis, University of Cambridge, October 1978.

[7] D R Cutting, D R Karger, J O Pedersen, and J W Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proc of the 15th Intl ACM SIGIR Conf*, pages 318–329, 1992.

[8] M Hemmje, C Kunkel, and A Willet. Lyberworld - a visualization user interface supporting fulltext retrieval. In *Proc of the 17th Intl ACM SIGIR Conf*, 1994.

[9] S Henninger and N J Belkin. Interface issues and interaction strategies for information retrieval systems. *CHI Electronic Proceedings*, 1996.

[10] R Korfhage. To see or not to see - is that the query? In *Proc of the 14th Intl ACM SIGIR Conf*, pages 134–141, 1991.

[11] P R Krishnaiah and L N Kanal. *Classification, Pattern Recognition and Reduction of Dimensionality*. North-Holland Publishing Company, 1982.

[12] A V Leouski and W B Croft. An evaluation of techniques for clustering search results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst, 1996.

[13] E Rasmussen. Clustering algorithms. In W B Frakes and R Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 419–442. Prentice Hall, 1992.

[14] S M Rüger. Validation of post-retrieval clustering. *submitted*, 2000.

[15] G Salton. *Automatic information organization and retrieval*. McGraw-Hill, New York, 1968.

[16] J W Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5), 1969.

[17] A Spoerry. Infocrystal: A visual tool for information retrieval & management. In *Proc of Information, Knowledge and Management 93*, 1993.

[18] C J van Rijsbergen. *Information Retrieval*. Butterworth, London, 2nd edition, 1979.

[19] E Voorhees. The cluster hypothesis revisited. In *Proc of ACM SIGIR*, pages 188–196, 1985.

[20] E M Voorhees and D K Harman. *Information Technology: The Seventh Text REtrieval Conf (TREC-7)*. NIST, 1999. http://trec.nist.gov.

[21] O Zamir and O Etzioni. Web document clustering: A feasibility demonstration. In *Proc of the 21th Intl ACM SIGIR Conf*, pages 46–54, 1998.

[22] G Zervas and S M Rüger. The curse of dimensionality and document clustering. In *Proc of the IEE Searching for Information: AI and IR Approaches*, 1999.

Acknowledgements: This work was partly supported by the EPSRC.

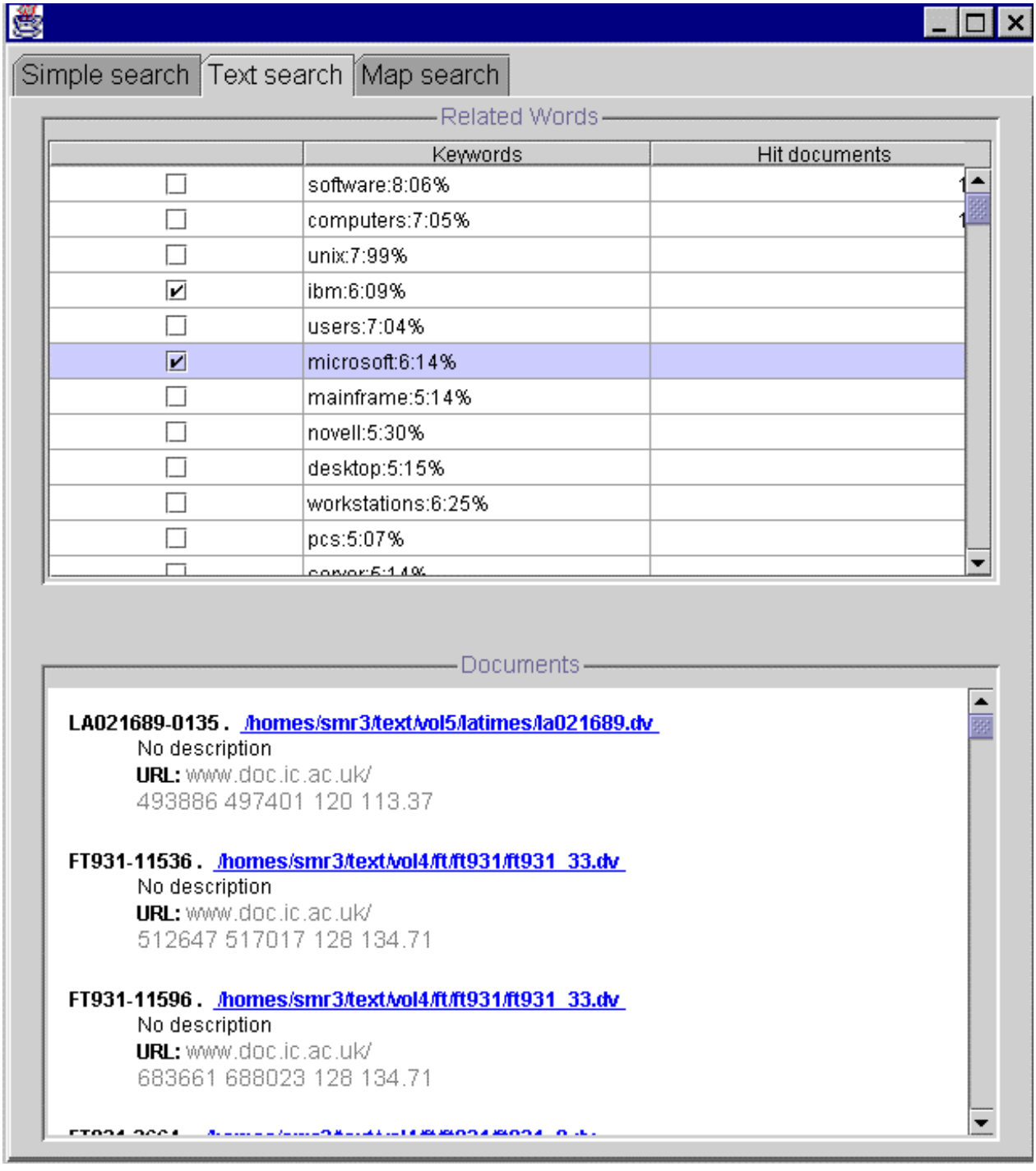

Figure 3: Panel for the keyword search

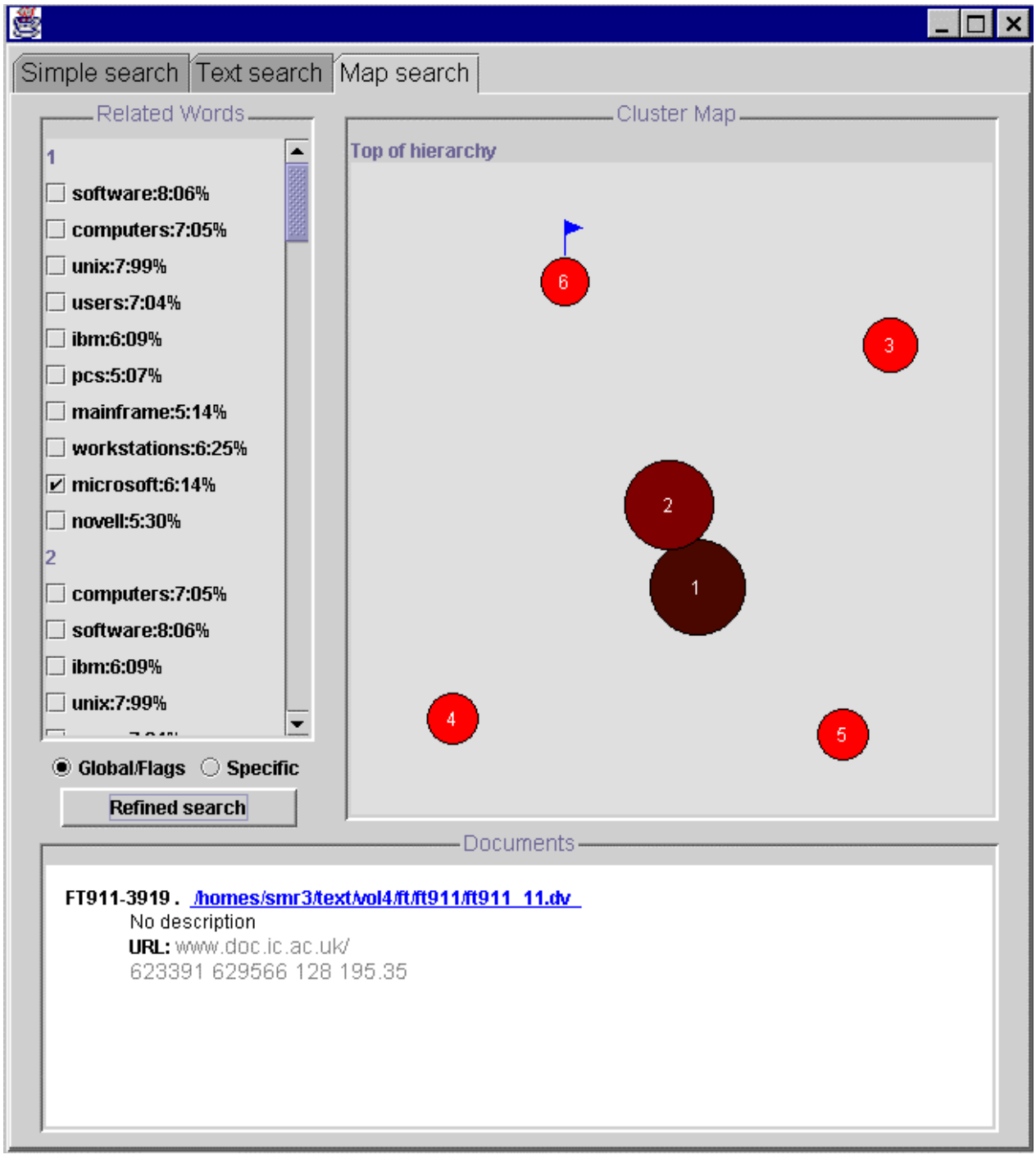

Figure 4: Panel for clustering-based search