Exploring web search results clustering

Xiaoxia Wang and Max Bramer

School of Computing, University of Portsmouth, UK

xiaoxia.wang@port.ac.uk, max.bramer@port.ac.uk

Abstract

As the number of documents on the web have proliferated, the low precision of conventional web search engines and the flat ranked list presentation make it difficult for users to locate specific information of interest. Grouping web search results into a hierarchy of topics provides an alternative to the flat ranked list and facilitates searching and browsing. In this paper, we have a brief survey of previous work on web search results clustering and existing commercial search engines using this technique, discuss two key issues of web search results clustering: cluster summarisation and evaluation, and propose the possibility of future research direction.

1. Introduction

Conventional web search engines (e.g. Google, MSN, AltaVista) often return a long list of ranked links in response to user queries. Web users have to go through the long ordered list of snippets extracted from original web pages to identify the information they are looking for. This problem gets worse as the number of documents on the web continues to grow, and especially when the user just wants to browse, has no specific searching aim or difficult to describe information needs by keywords.

There are some attempts using different methods to bring the large number of web documents into order:

The first method is pre-clustering (classification) of the entire corpus. It means clustering is performed in advance on the whole collection. For instance, web directories such as Yahoo! and LookSmart have been used to classify web pages [Chen and Dumais, 2000; Attardi and Sebastiani, 1999]. Although this approach has the advantage of using known and consistent category information to assist grouping similar documents, the manual nature of the directory makes it impossible to have as broad coverage as the popular search engines [Chen and Dumais, 2000] such as Google (www.google.com). Also, the predefined categories may not be useful in organizing the search results of a particular query. So it is no surprise that in Chen's experiment, results for some queries do not match any of the categories very well while results for others fall entirely within one category.

[Ferragina and Gulli, 2004] also pointed out that this method is not suitable in the web domain, where the results of queries could be extremely varied (in the number, length, type and relevance of the documents). They argued the web is a dynamic environment, any "fixed set of category labels would not be flexible enough" to capture the themes of web search results.

The second approach is post-retrieval document clustering. Document clustering has been explored by the IR (Information Retrieval) community as an alternative way of organizing retrieval results [Croft, 1978; Cutting et al., 1992; Allen et al., 1993; Leouski and Croft, 1996] and it suggests that clustering techniques can be used on the major search engines to help organize retrieved search results. The motivation of exploring this method is: clustering enables the user to discover patterns and structure in the document set that could be overlooked in the traditional ranked-list presentation [Cutting et al., 1992].

We investigate document clustering as a method that enables users to efficiently navigate through a large collection of search engine results. Document clustering algorithms attempt to group documents together based on their similarities; thus documents relating to a certain topic will hopefully be placed in the same cluster. This can help users both in locating interesting documents more easily and in getting an overview of the retrieved document set. Leouski and Allan (2000) tried to combine the ranked list and clustering presentation to improve the information retrieval efficiency.

Because the search results set is generated dynamically, the application of clustering on this kind of dynamic collections introduces new challenges to clustering technology. Zamir (1999) identified several key requirements of web search results clustering:

- Coherent clusters: the clustering algorithm should group similar documents together and also allow overlapping due to some documents having multiple topics.
- Ease-of-browsing: concise and accurate cluster description.
- Speed: as the clusters are generated for online use, it is crucial that the system does not introduce apparent delay to the search. So both *algorithmic speed* and *snippet-tolerance* are required.

This paper is a brief survey of web search results clustering, presenting and discussing the related research trying to achieve the above key requirements. The rest of this paper is organized as follows. Section 2 gives an overview of related work on web-based clustering techniques. Section 3 and section 4 discuss two key issues of ephemeral clustering that haven't been well addressed: search results clustering summarisation and evaluation. Section 5 concludes with pointing to future directions.

2. Related work

Scatter/Gather [Cutting et al., 1992; Hearst et al., 1996] is the first query result visualisation algorithm using the clustering technique in the Information Retrieval community. Before this work, document clustering was traditionally investigated mainly as a method for improving document search and retrieval, but was not widely used because of speed and poor performance of improving near-neighbour search. Instead of attempting to reduce the number of documents returned, Cutting et al. introduced document clustering as a document browsing method. They state that the Scatter/Gather system is particularly helpful in situations in which it is difficult or undesirable to specify a query formally: 1. when the user is not looking for anything specific, just wants to discover the general information content of the corpus (to gain an overview); 2. when it is difficult to formulate the query precisely (help user formulate a search request). Two near linear time clustering algorithms were presented: Buckshot and Fractionation. However, their work is based on general document collections, not on dynamically generated search results.

Zamir & Etzioni (1998) followed this paradigm and propose the notion of search results clustering (also called ephemeral clustering in [Maarek et al., 2000]). In their Grouper system, STC (Suffix Tree Clustering) treats a document as a string instead of a set of words. It attempted to cluster documents "snippets" returned by search engine according to common phrases they contain, thus employing information about the proximity and order of single keywords in addition to their frequencies.

STC is organized into two phases: discovering base clusters using a suffix tree and merging base clusters. In the first stage, the retrieved document "snippets" are inserted into a suffix tree, where each node in the tree represents a group of documents and a phrase that is common to all of them. For each phrase shared by two or more documents, they assign a score: $s(B)=|B| * f(|P|)$ to penalize single word terms, where $|B|$ is the number of documents in base cluster B, and $|P|$ is the length of the phrase P. Only the base clusters whose score is higher than an arbitrarily chosen minimal base cluster score are retained. In the second phase, N top-ranking base clusters are merged using a version of the AHC algorithm, with binary single-link merge criterion and predetermined minimal similarity between base clusters as the halting criterion [Zamir & Etzioni, 1999].

The two distinguishing features of STC are: linear time complexity; clustering documents according to shared phrases instead of word frequency. These make it "a substantial momentum" [Weiss and Stefanowski, 2001] of ephemeral clustering.

Carrot system built by Weiss and Stefanowski (2001) extended STC's application into the Polish Language, by using different stemming techniques. They investigate the influence of two primary STC parameters: merge threshold and minimum base cluster score on the number and quality of results produced by STC algorithm.

SHOC [Zhang and Dong, 2001] is based on latent semantic indexing and designed to work in Chinese. Two novel concepts are introduced to overcome STC's limitations: complete phrases and continuous cluster definition. A data structure called suffix array is used to identify complete phrases to avoid extracting meaningless partial phrases. Continuous cluster definition allows documents to be assigned to multiple clusters.

LINGO [Osinski, S., 2003] is a slightly modified version of the SHOC algorithm and is claimed as a “description oriented algorithm”. Being different from the previous approach, LINGO identifies cluster labels first using latent semantic indexing technique, retrieved search results are assigned to different groups based on the labels.

Wang and Kitsuregawa (2001) have proposed combining links and content in a k-means framework. The problem is that search engines don’t provide an easy access to the links graph, so the original web page needs to be downloaded and parsed to gather the ingoing/outgoing links information. This method is not feasible as the link analyse process can’t meet the speed requirement of online ephemeral clustering.

[Jiang et al., 2002] developed the Retriever system, comparing two different distance metrics: N-gram and Vector space model by clustering the data using a robust fuzzy relational algorithm. They compare the results with STC and find the N-gram based approach performs better than the vector space based approach, and as well as Zamir and Etzioni ’s (1998) STC algorithm. But the author pointed out that their search results are drawn from Lycos, whereas STC draws on Metacrawler. So the comparison between these two methods is arguable.

Microsoft [Zeng et al., 2004] proposed a system that reformalized the clustering problem as a salient phrase ranking problem. It extracts and ranks salient phrases from snippets based on a regression model, which is trained by human labelled data. Although this approach can benefit from labelled data, the additional training phase is hard to adapt to the Web [Ferragina and Gulli, 2005].

SnakeT [Ferragina and Gulli 2004, 2005] constructs two knowledge bases offline and takes advantage of them. The first one is called Anchor Text and Link Database. It is used to enrich the snippets returned by a search engine. The other is called Semantic Knowledge Base, it is used to help ranking in the process of generate “approximate sentences”. This approach is different from Grouper and other approaches as they treated sentences formed by contiguous terms, while SnakeT extracts sentences involving non-contiguous terms. It first extracts “approximate” sentences from the enriched snippets collection, then uses a knowledge base to help ranking, sentences above threshold are used as a set of meaningful labels. These labels will be then used to form and name the clusters and are called the primary label. To generate the labels of the nodes in the higher levels of the hierarchy, k-approximate sentences are used which have a good rank and occur in at least c% of the documents contained in the cluster. This set of secondary labels “provide a description for the cluster at a coarser level and thus is more useful for hierarchical formation and labelling”.

In addition to the above academic tools, there also has been a surge of commercial interest in novel IR-tools that support users in searching tasks [Ferragina and Gulli, 2004]. The following are existing industrial systems implementing clustering techniques in their (meta) search engines: Vivisimo, Grokker, Clusty and Iboogie provide cluster hierarchies in addition to the flat ranked list of search results;

Kartoo use a network visualisation interface, Mooter also use a network visualisation tool but followed by hierarchical clusters presentation when a node in the network is clicked; Copernic and Dog-pile concentrate more on supporting users on query formulation (providing revised/refined query suggestions). Among the various clustering search engines, Vivisimo.com deserves a special mention. This commercial meta search engine organises search results into hierarchical and very well described thematic groups and can be considered a benchmark and state-of-the-art in current research [Weiss, D., 2002, Osinski, S., 2003]. But very little information about this software is available as it is not publicly accessible. Much academic research attempts to address the search results clustering problem, but the attainable performance is far from the one achieved by Vivisimo. Only SnakeT claims to achieve efficiency and efficacy performance close to it [Ferragina and Gulli, 2005].

3. Clusters summarisation

Within the field of information retrieval, document clustering is also known as Automatic Taxonomy Generation (ATG): automatically build a hierarchical organization of words or phrases (terms) extracted from a set of documents to represent the concept of topics and subtopics in the cluster hierarchy. Krishnapuram and Kummamuru (2003) identified three main issues of ATG: how to identify documents that have similar content; how to discover the hierarchical structure of the topics and subtopics; how to find appropriate labels for each of the topics and subtopics. In this section, we focus on the third issue: how to generate labels for the hierarchical structure.

ATG algorithms can be categorized into different types depending on if the taxonomies are generated by clustering words or documents. There have been also some attempts at co-clustering, where documents and words are clustered at the same time.

3.1 ATG algorithms based on clustering words

The approaches based on clustering words focus on organizing words according to their thesaural relationship [Krishnapuram and Kummamuru, 2003]. There has been much research investigating automatically generating thesaural relationships from a corpus. Some is based on analysing the phrase in which a term occurs to infer the relationships between various terms [Grefenstette 1994, Hearst 1998]. Woods (1997) also uses phrasal analysis but in addition to a large knowledge base to organize terms into a concept hierarchy. A brief description of how they utilise different phrase analyse methods can be found in [Sanderson and Croft, 1999]. There are other methods using term co-occurrence to produce structure of related terms [Forsyth 1986; Sanderson and Croft 1999; Lawrie 2001; Kummamuru et al. 2001, 2004]. The details of these algorithms can be found in Krishnapuram's survey (2003). Because our focus is the application of ephemeral clustering of search results, in the following we briefly describe some of those methods that perform hierarchical clustering which is more suitable for browsing tasks.

The subsumption algorithm by Sanderson and Croft (1999) builds a concept hierarchy by finding pairs of concepts (m,n) in which m subsumes n. the algorithm computes the subsumption relationships between some selected pairs of words and phrases (m, n) from the document collection and then retains only those pairs in which m subsumes n. The hierarchy is then built in a bottom-up fashion. In this algorithm, the generality and specificity of terms was determined by their document frequency (DF), the more documents a term occurred in, the more general it was assumed to be.

Kummamuru et al. (2004) developed an algorithm called DisCover. It identified six desirable properties of taxonomies generated from a corpus of documents: document coverage, compactness, sibling node distinctiveness, node label predictiveness, reach time, general to specific. The first three properties are used for the development of the algorithms while the other three are only used in the evaluation phase. For the term extraction, the index is binary which only indicates whether a term occurs or does not occur in a document as Kummamuru argued that it makes no sense to consider frequency of term occurrence because only short snippets (1-3 lines) are being dealt with. At every level of hierarchy, this algorithm progressively identifies topic in a way that maximizes the coverage while maintaining distinctiveness of the topics. It was compared to two other monothetic clustering algorithms: DSP [Lawrie and Croft 2001], CAARD [Kummamuru 2001] and claimed that better hierarchies were generated according to user survey results.

From the above two examples, we can see the approaches based on clustering words first form a concept hierarchy by analysing the relationship between words, then assign documents (either soft or crisp) to appropriate nodes (topics and subtopics). These approaches are called monothetic clustering as the cluster assignment is based on a single feature. Monothetic clustering is claimed to be well suited for generating hierarchies for search results because “each cluster is described by a single feature or a concept and all the documents present in a cluster contain this feature” [Kummamuru et al. 2003] so that the user can easily understand clusters generated by monothetic clustering algorithms. But there have not been any experimental comparisons between monothetic and polythetic clustering of search results so far, maybe due to a lack of standard evaluation measures in this application area.

3.2 ATG algorithms based on clustering documents

The traditional clustering methods such as k-means and AHC (agglomerative hierarchical clustering) fall into this category. The basic idea is representing documents as N-dimensional vectors of word frequencies, where N is the total number of distinct non-stop words in the whole document collection. In the case of binary vectors, only the occurrence or non-occurrence of the word in the document is taken into account and recorded as 1 or 0. Once the documents are converted into vectors, appropriate similarity measures and clustering algorithms can be chosen for clustering. K-means clustering algorithm can be used to generate clusters in flat structure and AHC produce clusters of a bottom –up hierarchical tree. This approach to clustering documents is well studied in the literature and further details can be found in [Everitt, B., 1974; Anderberg, R., 1973]. The set of top ranking words with high frequency of occurrence within the cluster can be used as summarisation (label) for the cluster. There also have been attempts to use character n-grams to represent documents instead of using vector space method [Jiang, Z. et al., 2000].

We briefly described the STC algorithm by Zamir and Etzioni (1998) in section 2. This algorithm is different from traditional clustering methods since it is not based on vector space model. STC does not treat documents as a set of words but as a string. It captures shared phrases other than words as labels of the nodes of the hierarchy. But it needs to be noted that the phrases here are only contiguous terms due to the nature of the suffix tree.

The recently developed SnakeT system by Ferragina and Gulli (2005) attempts using a knowledge base to enrich the collection of words extracted from snippets to attain good labels of clusters. Approximate terms (involving non-contiguous terms) are extracted as hierarchy labels and attained a good performance.

From the above examples, we can see the approaches based on clustering documents form concept hierarchy by clustering documents first, then extract terms/sentence from documents in the cluster as meaningful labels. So it is a clusters-come-first approach and is also called polythetic clustering since the clusters are labelled by multiple concepts (terms).

3.3 Co-clustering methods

There is another approach called co-clustering, which clusters words and documents simultaneously. In section 3.2, we introduced that in traditional document clustering, documents are represented as N-dimensional vectors of word frequencies. Similarly, words can also be represented as M-dimensional vectors in which the i-th element represents the frequency of the word in i-th document. In this approach, the similarity between various words is computed according to their relative frequency distributions in the corpus [Pereira et al., 1993] so that words with similar document occurrence are grouped together; at the same time, the documents having similar keywords occurrence are grouped together**.** Thus, each co-cluster is actually a cluster of documents and an associated cluster of words [Mandhani, B. et al., 2003]. A set of selected words from each “word cluster” can be used as a label of the cluster.

The details of several co-clustering examples (FCoDoK, FSKWIC, RPSA) are covered by Krishnapuram et al.'s (2003) survey. The first two methods are similar to Fuzzy C-Means, thus generate clusters in flat structure. Only RPSA clusters documents hierarchically. Dhillon (2001) suggested a co-clustering algorithm based on bipartite graph partitioning. This method treats the document collection as a bipartite graph between words and documents and is finding the minimum cut vertex partitions in the graph. The top n words with greatest internal edge weights in each of the word clusters are selected to describe the associated document clusters.

4. Clustering evaluation

An important aspect of cluster analysis is the evaluation of clustering results. Halkidi, M. et al. (2001) made a comprehensive review of clustering validity measures available in the literature and classified them into three categories. In this section we briefly review the commonly used document clustering evaluation measures and the evaluation of search results clustering in the literature.

The first is the external evaluation method, which evaluates the results of clustering algorithm based on a pre-classified document set. There are several ways of comparing the clusters with the pre-defined classes: purity, entropy and mutual information. Purity can be computed by assigning class labels for each cluster by majority voting, then for a single cluster calculating the ratio of the correctly labelled documents to the total number of documents in the cluster. Entropy and mutual information take the entire distribution of documents into account instead of only considering the correct numbers of documents in the dominant class. They are defined in [Cover and Thomas, 1991]. If a cluster is viewed as the results of a query for a particular category, the F-measure [Van Rijsbergen, 1979] can also be used to evaluate the clustering results by combining precision and recall into a single number.

Because the search results are generated dynamically, there are no predefined categories to compare with. Sinka and Corne (2002) proposed a benchmark dataset for document clustering by extracting documents from Open Directory Projects (http://www.dmoz.org) and Yahoo! Categories (http://www.yahoo.com). But it is not suitable as ground truth under this paradigm as pre-defined category may not be suitable for search results of a particular query. One solution is to manually classify and assign labels to documents (e.g., Leouski and Croft, 1996) to make external evaluation possible. Zamir and Etzioni (1998) use an alternative method: manually assign a relevance judgement to each document based on user's information need (query's description), so that the clustering algorithm's effectiveness for information retrieval can be evaluated. Furthermore, Zamir and Etzioni (1999) compare the Grouper system interface to the traditional ranked-list interface through log analysis.

The second approach is based on internal criteria when ground truth is not available. Because the goal of clustering is to group a set of points into clusters so that points in the same cluster are more similar than points in different clusters [Jain and Dubes, 1988], the results of a clustering algorithm can be evaluated by using either minimized or maximized value of certain specified objective functions (e.g., the ratio of the average inter-cluster to intra-cluster distance). Berry and Linoff (1996) proposed two criteria: compactness and separation, which reflect the inter-cluster and intra-cluster similarity.

In addition to the above two criteria, we can also measure how well the labels in the hierarchy predict the contents of the clusters. Lawrie and Croft (2003) use Expected Mutual Information Measure (EMIM) [Van Rijsbergen, 1979] to compare the goodness of cluster labels with the top TF-IDF terms. Here we borrow the description of EMIM from Krishnapuram and Kummamuru (2003): Let T denote the set of label words in the hierarchy and let V denote the set of non-stop words that occur at least twice in the document collection. Then EMIM is given by

$$\text{EMIM (T, V)}=\sum_{t\in T}\sum_{v\in V} p(t,v)\log\frac{p(t,v)}{p(t)\,p(v)}$$

To test how well the hierarchy meets the end user requirements, Lawrie and Croft use another two metrics: one is called reachability, which measures the percentage of documents covered by the hierarchy; the other is to measure how quickly all the relevant documents can be found.

There are also other methods involved in evaluating clustering hierarchy quality. Sanderson and Croft (1999) performed a user study to judge the quality of relationship between child and its parent nodes in the hierarchy generated by subsumption, comparing with a random generated hierarchy. Lawrie and Croft (2000) also use parent-child pair in evaluation but they are only interested in how many common pairs are shared by two hierarchies, normalised by the total number of pairs in the hierarchy. The above two methods are classified as relative evaluation measures because only similarity between two hierarchies is of interest in this scenario.

The research from IBM [Kummamuru et al., 2004] combines the above measures to evaluate clustering hierarchy in terms of six desirable properties. They adopt *compactness* criteria from [Berry and Linoff, 1996] and interpret separation as *sibling node distinctiveness*, which is more suitable for hierarchy evaluation. The idea of using c*overage* and *reach time* metrics was originally from Lawrie and Croft (2000,2003), but the first metric was called reachability in their work. The *reach time* metric measures how quickly a user can reach all relevant documents by calculating the total nodes that must be traversed and the number of documents that must be read. *Node label predictiveness* and *general to specific* are difficult to quantify thus user study is used to rate the hierarchy of search results.

5. Summary and future research

In this paper, we present an overview of web search results clustering approaches and discuss two important aspects of web search results clustering: cluster summarisation and clustering evaluation. As the search results are retrieved from a (meta) search engine dynamically, it introduces lots of challenges and makes ephemeral clustering very different from traditional document clustering. There are some problems that have not been well addressed yet and we propose the possibilities of future research direction.

First, the goal of search results clustering is to provide an efficient searching and browsing tool for online use, thus raising the high requirement of speed and cluster summarization. The clustering is performed on snippets instead of full original web documents for efficiency. It implies that the cluster labels have to be extracted from short snippets (average 30-50 words) only. Obviously snippets are less informative than original web pages, thus making extracting cluster labels a challenging task. Some extract the most frequent keywords set as cluster label, some use continuous phrases, and recent research employ the approximate (gapped) phrases. SnakeT system [Ferragina and Gulli, 2005] exploits off-line knowledge base to enrich snippets collection so that gapped phrases with various length can be extracted as clusters label. This system was evaluated by user study and claimed the attainable performance is very close to Vivisimo (www.vivisimo.com), which is the state-of-the-art commercial clustering search engine. This led us to believe that using off-line information to aid clusters label extraction (especially on extracting gapped phrases) is a promising research direction.

Second, different approaches are proposed to automatically generate a hierarchy of search results and can be classified as document-based, term-based or co-clustering. Kummamuru et al. suggest classifying various approaches into monothetic clustering or polythetic clustering according to if the clustering is based on a single feature or multiple features. In our opinion, the first approach actually is to cluster documents first, then to extract cluster labels from the snippets under the cluster; while the second approach is to analyse the statistical distribution (e.g., co-occurrence, subsumption) of words in the whole collection and build a hierarchy first, then assign the documents to the nodes of the hierarchy. There has not been any comparison of these two methods yet may due to lacking of standard evaluation methods of search results clustering. In the future, standard evaluation methods need to be developed so that the monothetic clustering and polythetic clustering algorithms can be compared under a general framework.

Third, comparing and evaluating hierarchies is not easy as many of the criteria are subjective in nature. For example, the predictiveness of the cluster label and the relationship between parent and child nodes have to be evaluated by users. Therefore, user survey is widely used in search results clustering evaluation despite it being criticised as subjective and is limited in small scope evaluation (e.g., hard to involve large number of participants and hard to get large number of documents to be manually labelled). There are also attempts to measure user interface efficiency, but is criticised as misleading. To choose suitable objective and subjective measures and combine them to effectively evaluate search results clustering remains an open question.

References

Allen, R., Obry, P. and Littman, M. (1993) An interface for navigating clustered document sets returned by queries. In Proceedings of SIGOIS, pages 203--208, Malpitas CA, June 1993. ACM.

Anderberg, R. (1973) Cluster analysis for applications, London; New York: Academic Press.

Attardi, G., Gulli, A. and Sebastiani, F. Theseus (1999): categorization by context. In 8th International World Wide Web Conference.

Berry, A. and Linoff, G. (1997) Data Mining Techniques: For Marketing, Sales, and Customer Support, 1 ed., New York, USA, Wiley Computer Publishing.

Chen, H. and Dumais, S. (2000) Bringing Order to the Web: Automatically Categorizing Search Results. in Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems, pp. 142—152.

Cover, T. M., Thomas, J.A.: (1991) Elements of Information Theory. Wiley-Interscience.

Croft, W. B. (1978), Organising and Searching Large Files of Documents. PhD thesis, University of Cambridge.

Cutting, D. R., Karger, D. R., Pedersen, J. O. and Tukey, J. W. (1992) Scatter/Gather: a cluster-based approach to browsing large document collections. In proceedings of the 15th International ACM SIGIR Conference on research and development in information retrieval.

Dhillon, I. S. (2001) Co-Clustering Documents and Words Using Bipartite Spectral Graph Patitioning. KDD 2001, San Francisco, California, USA.

Everitt, B. Cluster Analysis (1974), London: Heinemann Educational (for) the Social Science Research Council.

Ferragina, P. and Gulli, A. (2004) The Anatomy of a Hierarchical Clustering Engine for Web-page, News and Book Snippets. Technical report, RR04-04 Informatica, Pisa.

Ferragina, P. and Gulli, A. (2005) A personalized Search Engine Based On Web-Snippet Hierarchical Clustering. In 14th International World Wide Web Conference.

Forsyth R. and Rada R. (1986) Adding an edge in machine learning: applications in expert systems and information retrieval, Ellis Horwood Ltd, pages 198-212.

Grefenstette, G. (1994) Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001) On Clustering Validation Techniques. Journal of Intelligent Systems, Vol 17:2/3, pages 107-145.

Hearst, M. A. and Pedersen, J. O. (1996) Re-examining the cluster hypothesis: Scatter/gather on retrieval results. In proceedings of SIGIR, pages 76-84, Zurich, CH, 1996.

Hearst, M.A. Automated discovery of WordNet relations. (1998) In Fellbaum, C., ed.: WordNet: an Electronic Lexical Database. MIT Press.

Jain, A. K. and Dubes, R. C. (1988) Algorithms for Clustering Data. Prentice Hall, New Jersey.

Jiang, Z. H., Joshi, A., Krishnapuram, R. and Yi, L. Y. (2002). Retriever: improving web search engine results using clustering. In Managing Business and Electronic Commerce.

Kummamuru, K. and Krishnapuram, R. (2001) A clustering algorithm for asymmetrically related data with its applications to text mining. In Proceedings of CIKM, pages 571--573, Atlanta, USA.

Krishnapuram, R. and Kummamuru, K. (2003) Automatic taxonomy generation: issues and possibilities. In LNCS: proceedings of fuzzy sets and systems (IFSA), volume 2715, pages 52-63. Spring-Verlag Heidelberg, Jan. 2003.

Kummamuru, K., Lotlikar, R. and Roy, S. (2004) A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. In SIGIR'04.

Lawrie, D. J. and Croft, W. B. (2000) Discovering and comparing topic hierarchies. In proceedings of RIAO 2000 conference, pages 314-330.

Lawrie, D. J., Croft, W. B. and Rosenberg, A. (2001) Finding topic words for hierarchical summarization. In Proceedings of SIGIR'01, pages 349-357. ACM press.

Lawrie, D. J. and Croft, W. B. (2003) Generating Hierarchical Summaries for Web Searches. SIGIR'03, Toronto, Canada.

Leouski, A. V. and Croft, W. B. (1996) An Evaluation of Techniques for Clustering Search Results. Technical report IR-76, Department of computer science, University of Massachusetts, Amherst.

Leuski A. and Allan J. (2000) Improving interactive retrieval by combining ranked list and clustering. Proceedings of RIAL, college de France, pp. 665-681, 2000.

Maarek, Y. S., Fagin, R., Ben-Shaul, I. Z. and Pelleg, D. (2000) Ephemeral Document Clustering for Web Applications. IBM research report RJ 10186.

Mandhani, B., Joshi, S. and Kummamuru, K. (2003) A Matrix Density Based Algorithm to Hierarchically Co-Cluster Documents and Words. In the International World Wide Web Conference 2003, Budapest, Hungary.

Osinski, S. (2003) An algorithm for clustering of web search results. Master Thesis. Poznan University of Technology, Poland.

Pereira, F., Tishby, N. and Lee, L. (1993), Distributional Clustering of English Words, In Meeting of the Association for Computational Linguistics, pp. 183-190.

Sanderson, M. and Croft, W. B. (1999) Deriving concept hierarchies from text, In proceedings of SIGIR, pages 206-213.

Sinka, M. P. and Corne, D. W. (2002): A large benchmark dataset for web document clustering, Soft Computing Systems: Design, Management and Applications, Vol 87, IOS Press, pp81-890.

Van Rijsbergen, C. J. (1979) Information retrieval, 2nd Edition, Department of Computer Science, University of Glasgow.

Wang, Y. and Kitsuregawa, M. (2001) Link based clustering of web search results. In proceedings of the 2nd International Conference on Web-Age Information Management (WAIM2001), Xi'An, P.R.China, Spring-Verlag LNCS, July, 2001.

Weiss, D. (2001). A clustering interface for web search results in Polish and English. Master Thesis, Pozna\'n University of Technology.

Weiss, D. and Stefanowski, J. (2003) Web search results in polish: experimental evaluation of Carrot. In Proceedings of the New Trends in Intelligent Information Processing and Web Mining Conference, Zakopane, Poland, 2003.

Woods, W. A. (1997) Conceptual Indexing: A better way to organize knowledge, a Sun Labs technical report: TR-97-61. Editor, Technical Reports, 901 San Antonio Road, Palo Alto, California 94303, USA.

Zamir O. and Etzioni, O., (1998) Web document clustering: a feasibility demonstration. SIGIR 98, Melbourne, Australia.

Zamir O. and Etzioni, O., (1999) Grouper: A dynamic clustering interface to web search results, in proceeding of the eighth international world wide web conference (WWW8), Toranto, Canada.

Zamir O. (1999) Clustering web documents: a phrase-based method for grouping search engine results, Doctoral dissertation, University of Washington.

Zeng, H. J., He, Q. C., Chen, Z., Ma, W. Y. and Ma, J. W. (2004) Learning to Cluster Web Search Results. SIGIR'04, Sheffield, South Yorkshire, UK.

Zhang, D. and Dong, Y. S. (2001). Semantic, Hierarchical, Online Clustering of Web Search Results. In ACM 3rd Workshop on Web Information and Data.