Volume 1 No.6, December 2011 ISSN 2224-3577
International Journal of Science and Technology

http://www.ejournalofsciences.org

An Analysis of Web Document Clustering Algorithms

[1]K. Sridevi, [2]R. Umarani, [3]V.Selvi
[1,3]Department of Computer Science,Nehru Memorial College,
Puthanampatti 621005, Trichy District,Tamilnadu, India.
[2]Sri Sarada College for Women,Salem,
Tamilnadu, India.

ABSTRACT

Evidently there is a tremendous increase in the amount of information found today on the largest shared information source, the World Wide Web. The process of finding relevant information on the web is overwhelming. Even with the presence of today's search engines that index the web it is difficult to wade through the large number of returned documents in a response to a user query. Furthermore, users without domain expertise are not familiar with the appropriate terminology thus not submitting the right query terms, leading to the retrieval of more irrelevant pages and the most relevant documents do not necessarily appear at the top of the query output sequence. Users of Web search engines are thus often forced to sift through the long ordered list of document "snippets" returned by the engines. This fact has lead to the need to organize a large set of documents into categories through clustering. The Information Retrieval community has explored document clustering as an alternative method of organizing retrieval results. Grouping similar documents together into clusters will help the users find relevant information quicker and will allow them to focus their search in the appropriate direction. Various web document clustering techniques are now being used to give meaningful search result on web. In this paper an analysis of the various categories of web document clustering and also the various existing web clustering engines with its relevant clustering techniques are presented.

Keywords: *Information Retrieval, Document Clustering, Search Results Clustering, Web Clustering Engines*

1. INTRODUCTION

Nowadays, the internet has become the largest data repository, facing the problem of information overload. In the same time, more and more people use the World Wide Web as their main source of information. The existence of an abundance of information, in combination with the dynamic and heterogeneous nature of the Web, makes information retrieval a tedious process for the average user. This has led to the need for the development of new techniques to assist users effectively navigate, trace and organize the available web documents, with the ultimate goal of finding those best matching their needs. One of the techniques that can play an important role towards the achievement of this objective is document clustering. The increasing importance of document clustering and the variety of its applications has led to the development of a wide range of algorithms with different quality.

Based on this model, the following are the key requirements for Web document clustering methods:

1. **Relevance**: The method ought to produce clusters that group documents relevant to the user's query separately from irrelevant ones.
2. **Browsable Summaries**: The user needs to determine at a glance whether a cluster's contents are of interest. Sifting through ranked lists is not replaced with sifting through clusters. Therefore the method has to provide concise and accurate descriptions of the clusters.
3. **Overlap**: Since documents have multiple topics, it is important to avoid confining each document to only one cluster.
4. **Snippet-tolerance**: The method ought to produce high quality clusters even when it only has access to the snippets returned by the search engines, as most users are unwilling to wait while the system downloads the original documents off the Web.
5. **Speed**: The clustering method ought to be able to cluster up to one thousand snippets in a few seconds. For the impatient user, each second counts.
6. **Incrementality**: To save time, the method should start to process each snippet as soon as it is received over the Web.

This paper gives an idea about web document clustering, the various categories of clustering and analysis of existing clustering engines with the algorithms it supports.

2. DOCUMENT CLUSTERING

Clustering (or cluster analysis) is one of the main data analysis techniques and deals with the organization of a set of objects in a multidimensional space into unified groups, called clusters. Each cluster contains objects that are very similar to each other and very dissimilar to objects in other clusters. Cluster analysis aims at discovering objects that have some representative behavior in the collection. Clustering is a

form of unsupervised classification, which means that the categories into which the collection must be partitioned are not known. In order to cluster documents, one must first choose the type of the characteristics or attributes (e.g. words, phrases or links) of the documents on which the clustering algorithm will be based and their representation. The most commonly used model is the Vector Space Model. Vector Space Model is a mathematical model to represent Information Retrieval Systems which uses term sets to represent both documents and queries, employs basic linear algebra operations to calculate global similarities between them.

3. AN ANALYSIS ON CATEGORIES OF WEB DOCUMENT CLUSTERING

There are many document clustering approaches available [1]. They differ in many parts, such as the types of attributes they use to characterize the documents, the similarity measure used, the representation of the clusters etc. Based on the characteristics or attributes of the documents that are used by the clustering algorithm, the different approaches can be categorized into *i.* ***text-based***, in which the clustering is based on the content of the document, *ii.* ***link-based***, based on the link structure of the pages in the collection and *iii.* ***hybrid*** ones, which take into account both the content and the links of the document.

3.2 Text based Clustering

The text-based web document clustering approaches characterise each document according to its content, i.e. the words (or sometimes phrases) contained in it. The basic idea is that if two documents contain many common words then it is likely that the two documents are very similar.

The text-based approaches can be further classified according to the clustering method used into the following categories: ***flat/partitional***, ***hierarchical***, ***graph- based***, ***neural network-based*** and ***probabilistic***. Furthermore, according to the way a clustering algorithm handles uncertainty in terms of cluster overlapping. An algorithm can be either *crisp* (or hard), which considers non-overlapping partitions, *fuzzy* (or soft), with which a document can be classified to more than one cluster. Most of the existing algorithms are crisp, meaning that a document either belongs to a cluster or not. It must also be noted that most of the mentioned approaches in this category are general clustering algorithms that can be applied to any kind of data.

3.2 Partitional Clustering

The partitional or non-hierarchical document clustering approaches attempt a flat partitioning of a collection of documents into a *predefined* number of *disjoint* clusters. The most known class of partitional clustering algorithms are the k-means algorithm and its variants. K-means algorithms are O(nkT), where T is the number of iterations, which is considered more or less a good bound. However, a major disadvantage of k-means is that it assumes spherical cluster structure, and cannot be applied in domains where cluster structures are non-spherical. A variant of k-means that allows overlapping of clusters is known as Fuzzy C-means (FCM). Instead of having binary membership of objects to their respective clusters, FCM allows for varying degrees of object memberships. Krishnapuram et al proposed a modified version of FCM called Fuzzy C-Medoids (FCM) where the means are replaced with medoids. Due to the random choice of cluster seeds these algorithms exhibit, they are considered non-deterministic as opposed to hierarchical clustering approaches. One approach that combines both partitional clustering with hybrid clustering is the bisecting k-means algorithm.

3.3 Hierarchical Clustering

Hierarchical techniques produce a nested sequence of partitions, clusters at an intermediate level encompass all the clusters below them in the hierarchy. The result of a hierarchical clustering algorithm can be viewed as a tree, called a dendogram. The dendrogram is a useful representation when considering retrieval from a clustered set of documents, since it indicates the paths that the retrieval precess may follow. Depending on the direction of building the hierarchy, the following two kinds of hierarchical clustering are possible: Agglomerative and Divisive. The agglomerative approach is the most commonly used in hierarchical clustering.

Agglomerative Hierarchical Clustering (AHC): This method starts with the set of objects as individual clusters, then at each step merges the most two similar clusters. This process is repeated until a minimal number of clusters have been reached, or, if a complete hierarchy is required then the process continues until only one cluster is left. This method is very simple but needs to specify how to compute the distance between two clusters. Three commonly used methods for computing this distance are listed below:

Single Linkage Method: The similarity between a pair of clusters is the maximum of the similarities between all pairs of documents such that one document is in one cluster and the other document is in other cluster. This method is also called “nearest neighbour” clustering method.

Complete Linkage Method: The similarity between a pair of clusters is calculated as the minimum of the similarities between all pairs of documents. This method is also called “furthest neighbour” clustering method.

Average Linkage Method: This method produces clusters such that each document in a cluster has greater average similarity with the other documents in the cluster than with the documents in any other cluster. This method takes into account all possible pairs of distances between the objects in the clusters, and is considered more reliable and robust to outliers. This method is also known as UPGMA (Unweighted PairGroup Method using Arithmetic averages).

3.4 Divisive Hierarchical Clustering

These methods work from top to bottom, starting with the whole data set as one cluster, and at each step split a cluster until only singleton clusters of individual objects remain. One method is to find the two sub-clusters using k-means, resulting in a hybrid technique called bisecting k-means. Another method based on statistical approach is used by the ITERATE algorithm, however, it does not necessarily split the cluster into only two clusters, the cluster could be split up to many sub-clusters according to a cohesion measure of the resulting sub-partition.

3.5 Graph based Clustering

In this case the documents to be clustered can be viewed as a set of nodes and the edges between the nodes represent the relationship between them. The edges bare a weight, which denotes the strength of that relationship. Chameleon's graph representation of the document set is based on the k- nearest neighbor graph approach. Another graph based approach is the algorithm proposed by Dhillon which uses iterative bipartite graph partitioning to co-cluster documents and words. The advantages of these approaches are that can capture the structure of the data and that they work effectively in high dimensional spaces. The disadvantage is that the graph must fit the memory.

3.6 Neural Network based Clustering

The Kohonen's Self-Organizing feature Maps (SOM) is a widely used unsupervised neural network model. Another approach proposed in the literature is the *hierarchical feature map* model, which is based on a hierarchical organization of more than one self-organizing feature maps. The aim of this approach is to overcome the limitations imposed by the 2-dimensional output grid of the SOM model, by arranging a number of SOMs in a hierarchy, such that for each unit on one level of the hierarchy a 2-dimensional self-organizing map is added to the next level. Neural networks are usually useful in environments where there is a lot of noise, and when dealing with data with complex internal structure and frequent changes. The advantage of this approach is the ability to give high quality results without having high computational complexity. The disadvantages are the difficulty to explain the results and the fact that the 2-dimensional output grid may restrict the mirroring and result in loss of information.

3.7 Fuzzy Clustering

All the aforementioned approaches produce clusters in such a way that each document is assigned to one and only one cluster. Fuzzy clustering approaches, on the other hand, are non-exclusive, in the sense that each document can belong to more than one clusters. The most widely used fuzzy clustering algorithm is Fuzzy c-means, a variation of the partitional k-means algorithm. Another fuzzy approach, that tries to overcome the fact that fuzzy c- means doesn't take into account the distribution of the document vectors in each cluster, is the Fuzzy Clustering and Fuzzy Merging algorithm (FCFM). The FCFM uses Gaussian weighted feature vectors to represent the cluster prototypes.

3.8 Probabilistic Clustering

Another way of dealing with uncertainty is to use probabilistic clustering algorithms. These algorithms use statistical models to calculate the similarity between the data instead of some predefined measures. Two widely used probabilistic algorithms are Expectation Maximization (EM) and AutoClass. The output of the probabilistic algorithms is the set of distribution function parameter values and the probability of membership of each document to each cluster.

3.9 Using Ontologies

The categories of clustering methods described above, most often rely on *exact* keyword matching, and do not take into account the fact that the keywords may have some *semantic proximity* between each other. THESUS is a system that clusters web documents that are characterized by weighted keywords of an ontology. The ontology used is a tree of terms connected according to the IS-A relationship. Firstly, the keywords that characterize each document are mapped onto terms in the ontology. Then, the similarity between the documents is calculated based on the proximity of their terms in the ontology. Finally, a modified version of the DBSCAN clustering algorithm is used to provide the clusters. The advantage of using an ontology in clustering is that it provides a very useful structure not only for the calculation of document similarity, but also for dimensionality reduction by abstracting the keywords that characterize the documents to terms in the ontology.

4. EXISTING WEB CLUSTERING ENGINES

[8] has given a particular analysis on comparison of various clustering engines.

Table 1: Comparison of Clustering Engines

Name	Algorithm	Clustering	Time Complexity
Grouper	STC	Flat	O(n)
Carrot	Lingo	Flat	O(n)
Vivisimo	Lingo	Hierarchical	O(n)
iBoogie	STC	Hierarchical	O(n)
Web Cat	K Means	Flat	O(nkt)
WICE	SHOC	Hierarchical	O(n)

4.1 Grouper

Grouper [3] is a document clustering interface to the HuskySearch meta-search service. HuskySearch (which is based on MetaCrawler [9]) retrieves results from several popular Web search engines, and Grouper clusters the results as they arrive using the STC algorithm.

Figure 1: Grouper: A Document Clustering Interface

MetaCrawler [9] is a metasearch engine that blends the top web search results from Google, Yahoo!, Bing (formerly Live Search), Ask.com, About.com, MIVA, LookSmart and other popular search engines. MetaCrawler also provides users the option to search for images, video, news, yellow pages and white pages. It used to provide the option to search for audio.

URL: http://www.metacrawler.com/

Figure 2: Metacrawler: A Meta Search Engine

4.2 Carrot2

Carrot2 [10] [14] combines several search results clustering algorithms: STC, Lingo, TRSC, clustering based on swarm intelligence (ant-colonies), and simple agglomerative Techniques. Lingo uses SVD as the primary mechanism for cluster label induction.

URL: http://search.carrot2.org/stable/search

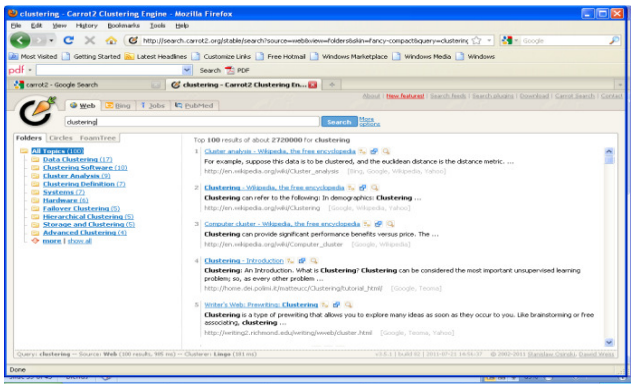

Figure 3: Carrot2: A Web Clustering Engine

4.3 Vivisimo

Vivísimo [11] was founded by research computer scientists at the Computer Science Department at Carnegie Mellon University, where research was originally done under grants from the National Science Foundation. The company was founded in June 2000. This algorithm is based on an old artificial intelligence idea, a good cluster or document grouping is one, which possesses a good, readable description. Their document clustering and meta-search software automatically categorizes search results on-the-fly into hierarchical clusters. Vivisimo Velocity is built on a modern architecture and takes advantage of XML and XSL statndards.

URL: http://vivisimo.com/resources/demos.html

Figure 4: Vivisimo: A Web Clustering Engine

ISSN 2224-3577

4.4 iBoogie

iBoogie [12] is a search site developed and owned by CyberTavern. IBoogie combines metasearch and clustering to deliver and organize search results from multiple sources into structured content.

URL: http://www.iboogie.com/

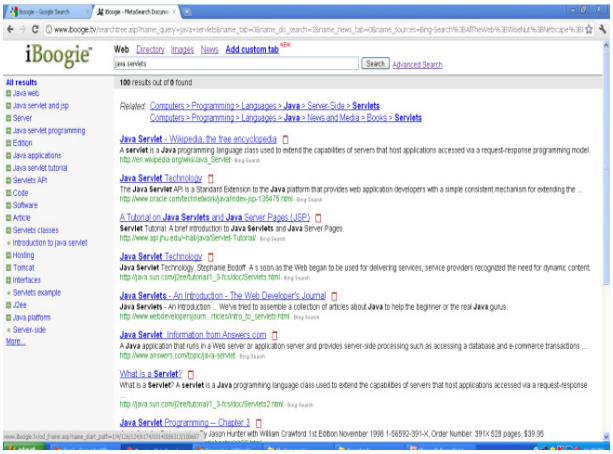

Figure 5: iBoogie: A Web Document Clustering Engine

4.5 WICE (SHOC)

WICE (Web Information Clustering Engine) devise an algorithm called SHOC (Semantic Hierarchical Online Clustering) that handles data locality and successfully deals with large alphabets. For solving the first problem, SHOC uses suffix arrays instead of suffix trees for extracting frequent phrases.

4.6 WebCAT

WebCAT was built around an algorithm for clustering categorical data called *transactional k-Means*. This is originally developed for databases, this algorithm has little to do with transactional processing and is rather about careful definition of (dis)similarity (Jaccard coefficient) between objects (documents) described by categorical features (words). WebCAT's computational complexity is linear in the number of documents to be clustered, assuming a fixed number of iterations.

5. ALGORITHMS USED IN WEB DOCUMENT CLUSTERING

5.1 Agglomerative Hierarchical Clustering (AHC) Algorithm

Numerous documents clustering algorithms are available. Agglomerative Hierarchical Clustering (AHC) algorithms are probably the most commonly used. These algorithms are typically slow when applied to large document collections. These algorithms are very sensitive to the halting criterion when the algorithm mistakenly merges multiple "good" clusters, the resulting cluster could be meaningless to the user. In the Web domain where the results of queries could be extremely varied (in the number, length, type and relevance of the documents), this sensitivity to the halting criterion often causes poor results. Another characteristic of the Web domain is that we often receive many outliers. This sort of "noise" reduces even further the effectiveness of commonly used halting criteria. Buckshot and Fractionation are fast, linear time clustering algorithms introduced in. Buckshot is a K-Means algorithm where the initial cluster centroids are created by applying AHC clustering. Fractionation is an approximation to AHC, where the search for the two closest clusters is not performed globally, but in rather locally and in a bound region. This algorithm will obviously suffer from the same disadvantages of AHC namely the arbitrary halting criteria and the poor performance in domains with many outliers.

5.2 K-Means Algorithm

Linear time clustering algorithms are the best candidates to comply with the speed requirement of on-line clustering. These include the K-Means algorithm and the Single-Pass method. One advantage of the K-Means algorithm is that, unlike AHC algorithms, it can produce overlapping clusters. Its chief disadvantage is that it is known to be most effective when the desired clusters are approximately spherical with respect to the similarity measure used. There is no reason to believe that documents should fall into approximately spherical clusters. The Single-Pass method also suffers from this disadvantage, as well as from being order dependant and from having a tendency to produce large clusters. It is, however, the most popular incremental clustering algorithm.

5.3 Suffix Tree Clustering (STC)

To satisfy the stringent requirements of the Web domain, an incremental, linear time (in the document collection size) algorithm called Suffix Tree Clustering (STC) [2] [7] is introduced, which creates clusters based on phrases shared between documents. It is shown that STC is faster than standard clustering methods in this domain, and argue that Web document clustering via STC is both feasible and potentially beneficial. STC does not treat a document as a set of words but rather as a string, making use of proximity information between words. STC relies on a suffix tree to efficiently identify sets of documents that share common phrases and uses this information to create clusters and to succinctly summarize their contents for users.

Strong and weak points of STC

A clear advantage of Suffix Tree Clustering is that it uses phrases to provide concise and meaningful descriptions of groups. However, STC's thresholds play a significant role in the process of cluster formation, and they turn out particularly difficult to tune. Also, STC's phrase pruning heuristic tends to remove longer high quality phrases, leaving only the shorter and less informative ones. Finally, if a document does not include any of the extracted phrases or just some parts of them, it will not be included in the results although it may still be relevant.

5.4 Semantic Hierarchical Online Clustering (SHOC)

The Semantic Online Hierarchical Clustering [5] is a web search results clustering algorithm that was originally designed to process queries in Chinese. Although it is based on a variation of the Vector Space Model called Latent Semantic Indexing (LSI) and uses phrases in the process of clustering, it is much different from the its predecessors. To overcome the STC's low quality phrases problem, An algorithm is proposed that uses a data structure called suffix array to identify complete phrases and their frequencies in O(n) time, n being the total length of all processed documents. The SHOC algorithm works in three main phases: complete phrase discovery phase, base cluster discovery phase and cluster merging phase.

Weak points of SHOC

One of the drawbacks of SHOC is that Zhang and Dong provide only vague comments on the values of thresholds of their algorithm and the method which is used to label the resulting clusters.

It also shows that in many cases the Singular Value Decomposition produces unintuitive, sometimes even close to "random", continuous clusters. The reason for this lies probably the fact that the SVD is performed on document snippets rather than the full texts as it was in its original applications.

5.5 The Description-Comes-First Approach (LINGO)

As pointed out by [4] [5] [6] , Lingo describes the search results clustering method on which the main idea behind the description comes first approach is that the process of clustering is *reversed*: it is first found the meaningful cluster labels and only then assign snippets to them to create proper groups.

Lingo and SHOC

Table 2: Comparison of Lingo with SHOC

Topic	SHOC	Lingo
Phase Identification	Based on Suffix Arrays	Adapted from SHOC
Label Discovery	Performed after cluster content discovery	Based on Singular Value Decomposition, performed before cluster content discovery
Cluster content discovery	Based on Singular Value Decomposition	Based on cluster labels, employs the Vector Space Model
Post processing	Hierarchical cluster merging based on content overlap	No cluster merging applied

Differences between Lingo and other algorithms

The main difference between Lingo and other search results clustering algorithms is in the way they try to explain the property that makes snippets in one cluster similar to each other. In previous approaches documents were assigned to groups according to some abstract mathematical properties, which would sometimes lead these algorithms down a blind alley of *knowing* that certain documents should be clustered together and at the same time being unable to *explain* the relationship between them in a human-readable fashion. In the description-comes-first approach this problem is avoided by *first* finding readable descriptions and only then trying to create appropriate clusters.

6. Comparison of Web Clustering Algorithms

Table 3 Comparison of various Algorithms supported by diverse Web Clustering Engines

Algorithm	Time Complexity	Advantages	Disadvantages
Agglomerative Hierarchical Clustering (AHC)	Single link and group average: $O(n^2)$ Complete link: $O(n^3)$	-Simple.	-Slow when applied to large document collections. -Sensitive to halting criterion. -Poor performance in domains with many outliers.

K – means	O(nkt) (k:initial clusters, t: iterations)	-Efficient and simple. -Suitable for large datasets.	-Very sensitive to input parameters.
Suffix Tree Clustering (STC)	O(n)	- Incremental -Uses phrases to provide concise and meaningful description of groups.	-Snippets usually introduce noise. -Snippets may not be a good description of a web page.
Semantic Online Hierarchical Clustering (SHOC)	O(*n*)	-Uses Latent Semantic Indexing (LSI) and phrases in the process of clustering. -Uses suffix array to identify complete phrases. -Allows overlapping clusters. -Provides a method of ordering documents *within* clusters.	-Provides only vague comments on the values of thresholds of the algorithm and the method which is used to label the resulting clusters.
Lingo	O(n)	-Readable cluster labels. - Overlapping clusters. -Cluster accuracy.	-Unable to generate a hierarchical structure of clusters. -The implementation of lingo is fairly computationally expensive.

In the above, Agglomerative Hierarchical Clustering & K- means algorithms are crisp clusters and Suffix Tree Clustering, Semantic Online Hierarchical Clustering & Lingo are of fuzzy clusters.

7. CONCLUSION

Clustering can increase the efficiency and the effectiveness of information retrieval. The fact that the user's query is not matched against each document separately, but against each cluster can lead to an increase in the effectiveness, as well as the efficiency, by returning more relevant and less non relevant documents. The organization and presentation of the pages in small and meaningful groups (usually followed by short descriptions or summaries of the contents of each group) gives the user the possibility to focus exactly on the subject of his interest and find the desired documents more quickly. Thus document clustering is very useful to retrieve information application in order to reduce the consuming time and get high precision and recall. This paper has presented an analysis of various algorithms that support web document clustering.

REFERENCES

[1] Oikonomakou, Nora, and Michalis Vazirgiannis. "A Review of Web Document Clustering Approaches." Data Mining and Knowledge Discovery Hankbook.

[2] O. Zamir and O.Etzioni, "Web Document Clustering: A Feasibility Demonstration", Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, pp. 46-54, 1998.

[3] O. Zamir and O.Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results", Computer Networks, Amsterdam, Netherlands, 31(11-16), pp. 1361-1374, 1999.

[4] Stanislaw Osinski, Jerzy Stefanowski and Dawid Weiss, "Lingo: Search Results Clustering Algorithm Bases on Singular Value Decomposition", Institute of Computing Science, Poznan University of Technology, 2004.

[5] Stanislaw Osinski, "Dimensionality Reduction Techniques for Search Results clustering", Master Thesis, Department of Computer Science, The University of Sheffield, UK, 2004.

[6] Dell Zhang and Yisheng Dong, "Semantic, Hierarchical, Online Clustering of Web Search Results", Proceeding of the 6th of Asia Pacific Web Conference (APWEB), Hangzhou, China, April 2004.

[7] Hung Chim, Xiaotie Deng, "A New Suffix Tree Similarity Measure for Document Clustering", Proceedings of the 16th International Conference on World Wide Web (WWW), Banff, Alberta, Canada, pp. 121-130, 8-12 May, 2007.

[8] R. Subhashini and V. Jawahar Senthil Kumar, "The Anatomy of Web Search Result Clustering

and Search Engines", Indian Journal of Computer Science and Engineering, Vol. 1, No. 4, pp. 392-401, 2006.

[9] http://www.metacrawler.com

[10] http://search.carrot2.org/stable/search

[11] http://vivisimo.com/resources/demos.html

[12] http://www.iboogie.com

[13] http://www.google.com

[14] http://demo.carrot-search.com