DRTC Workshop on
Semantic Web
8th – 10th December, 2003
DRTC, Bangalore

Paper: L

Organisation of Web pages Using Document Clustering Techniques: Some Algorithms

Bibhuti Bhusan Sahoo
Documentation Research and Training Centre
Indian Statistical Institute
Bangalore-560 059
bibhuti@drtc.isibang.ac.in

Abstract

With the increase in information on the World Wide Web it has become difficult to find the desired information on search engines. Clustering Techniques are now being used to give a meaningful search result on web. This paper gives an idea about Web Page document clustering, different algorithms including similarity measures and clustering engines.

1. Introduction

Clustering, a mostly used technique in data mining identifies a group of related records that can be used as a starting point for exploring further relationship among the data sets. The exponential growth of information on the World Wide Web has prompted for developing efficient and effective methods for organizing and retrieving the information. Clustering techniques play an important role in searching and organization of web pages. Traditional IR approaches are hardly appropriate in the context of the web, due to both the enormous size and hyper linked nature of the web. The majority of search engines give a long list of ranked documents; most of them are irrelevant. The low precision of the web search engines coupled with the long ranked list presentation make it hard for users to find the information they are looking for. Typical queries retrieve hundreds of documents, most of which have no relation with what the user was looking for. The limitations of search technology can be attributed to the following:

Polysemy: the words involved in the search have multiple meanings. For example, a user searching for windows may be interested in either the operating system or the physical artifact.

Phrases: a phrase may be different from words in it. e.g., the meaning of the phrase "partition magic" (a disk partition management tool) is quite different from the meaning of the individual words "partition" and "magic".

Term dependency: words in the terms are not totally independent of each other. For example, a user may look for details about a product made by a particular company and type in Sun's Enterprise Computer Series. Obviously, each word in this term is dependent on each other (5).

These problems are independent of how good the algorithms that associate keywords with the contents of a page are. One possible solution to this problem is to realize that the responses from search engines to a particular query can be broadly grouped into meaningful categories. If the user is shown these groups, possibly with some keyword type descriptions, they can then of course select one (or more) that fit their perceived interests. This is different from the site oriented grouping that some search engines present.

Clustering algorithms attempt to group documents together based on their similarities; thus documents relating to a certain topic will hopefully be placed in a single cluster. This can help users both in locating interesting documents more easily and in getting an overview of the retrieved document set. Several researchers have suggested that the clustering techniques are feasible for web mining. In this paper, an attempt has been made to discuss few methods for web page clustering

2. Key Requirements for Web Document Clustering

As pointed out by Zamir and Etzioni (4) the followings 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.

2. **Browsable Summaries:** The user needs to determine at a glance whether a cluster's contents are of interest. Ranked lists of the clusters may infact difficult to browse. 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**: A very patient user might sift through 100 documents in a ranked list presentation. Clustering on the other hand allows the user to browse several related documents. Therefore 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.

3. Previous Work on Document Clustering

Several researchers have been studying the document clustering and Numerous documents clustering algorithms appear in the literature. *Agglomerative Hierarchical Clustering* (AHC) algorithms are probably the most commonly used. These algorithms are typically slow when applied to large document collections. It is too slow to meet the speed requirement for one thousand documents. K-Means clustering algorithms are the best candidates to comply with the speed requirement of on-line clustering. These include $O(nkT)$ time complexity where k is the number of desired clusters and T is the number of iterations (Rocchio,), and the Single-Pass method - $O(nK)$ were K is the number of clusters created (Hill). 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.

The Buckshot and Fractionation are fast, linear time clustering algorithms introduced in (Cutting et. al., 92). The 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. The Buckshot is a K-Means algorithm where the initial cluster centroids are created by applying AHC clustering to a sample of the documents of the collection. In contrast to STC, all the mentioned algorithms treat a document as a set of words and not as an ordered sequence of words, thus losing valuable information. Phrases have long been used to supplement word-based indexing in IR systems (*e.g.*, Buckley et. al.). The

use of lexical atoms and of syntactic phrases has been shown to improve precision without hurting recall (Zhai et. al., 95). Phrases generated by simple statistical approaches (e.g., contiguous non-stopped words) have also been successfully used (Salton et. al, 75; Fagan, 87; Hull et. al., 97).

On the Internet, few attempts have been made to handle the large number of documents returned by search engines. Many search engines provide query refinement features. AltaVista, for example, suggests words to be added or to be excluded from the query. These words are organized into groups, but these groups do not represent clusters of documents. The Northern Light search engine (*www.nlsearch.com*), provides "Custom Search Folders", in which the retrieved documents are organized. Each folder is labeled by a single word or a two-word phrase, and is comprised of all the documents containing the label.

4. Similarity Metric

Clustering objects into subgroups is usually based on a similarity metric between objects, with the goal that objects within a subgroup are very similar, and objects between different subgroups are not similar. In clustering of a graph, similarity between nodes is represented as weight of the edge. In web page clustering problem, the followings can be incorporated like link structure, text information, and co-citation information into the similarity metric.

4.1. Hyperlink Structure

The link information is obtained directly from the link graph. Link structure alone provides rich information on the topic. By exploring the link structure, one can able to extract useful information from the web . One of the most popular algorithms to retrieve information from the link structure is Kleinberg's HITS algorithm, which will be discussed in section 5.1.

4.2. Textual information

The textual information is often included for clustering the web pages. Moreover, unlike printed literature, web text references each other more randomly. One approach that textual information has been incorporated is to measure the similarity between user query and the anchor text (text between (A HREF=...) and (</A>) . He et.al. experimented with this approach and found it did not work as effective in their data sets. So they used a new approach that

(a) utilizes the entire text of a web page, not just the anchored words;

(b) measures textual similarity S_{ij} between two web pages i; j, instead of between user query and the web page.

(c) use S_{ij} as the strength of the hyperlink between web pages i; j. The key observation here is that if two web pages have very little text similarity, it is unlikely that they belong to the same topic, even though they are connected by a

hyperlink. Therefore S_{ij} properly gauges the extent or the importance of an individual hyperlink.

They assumed each web page a document. The content of a document is obtained using a web crawler written in Perl. To accommodate the vast differences in web page lengths, they only use the first 500 words of each document; the rest of the document is discarded if it has more than 500 words. After text of all web pages is preprocessed, they represented each web page by a vector in the vector space model of IR. For each element of the vector, the standard tf.idf weighting is used: tf(i;j)*idf(i). tf(i; j) is the term frequency of word i in document j, representing the number of occurrence of word i in document j. idf(i)is the Inverse Document Frequency for word i, computed as

$$idf(i) = \log \left\{ \frac{\text{no. of total docs}}{\text{no. of docs containing word i}} \right\}$$

They computed the similarity (or relevance) between two web pages using the standard cosine similarity measure. If x and y are vectors of two documents j_1 and j_2, the Similarity between j_1 and j_2 is:

$$S(j_1 ; j_2) = \frac{\Sigma i \; x(i)*y(i)}{\sqrt{||x||^2 * ||y||^2}}$$

This simple textual similarity was the starting point of their approach ;

4.3. Co-citation

Co-citation is yet another metric to measure the relevance of two web pages. If there are many pages pointing to both of them, then these two pages are likely to address the similar issue. The co-citation C (i; j) of pages i and j is the number of web pages pointing to both i and j. The co-citation matrix C is easily obtained from the link graph.

The overall similarity between two web pages is the combination of above three factors.

5. Web Page Clustering Algorithms

Ranking web pages using the information contained in the hyperlinks between web pages currently is an active research area. Two popular ranking methods are PageRank of Brin and Page, and the HITS algorithm of Kleinberg . Another algorithm, which is developed by Zamir and Etzioni, is known as Suffix Tree Clustering (STC) . These are discussed below.

5.1. HITS Algorithm

The HITS (hyperlink-induced topic search) algorithm was first introduced by Jon M. Kleinberg (4) in 1998. He assumes that a topic can be roughly divided into pages with good coverage of the topic, called authorities, and directory-like pages with many hyperlinks to useful pages on the topic, called hubs. And the goal of HITS is basically to identify good authorities and hubs for a certain topic, which is usually defined by the user's query. So, HITS is a query-based algorithm.

Given a user query, the HITS algorithm first creates a neighborhood graph for the query. The neighborhood contained top 200 matched web pages retrieved from a content-based web search engine; it also contains all the pages of the 200 web pages linked to and pages that linked to these 200 top pages.

Then, an iterative calculation was performed on the value of authority and value of hub. For each page p , the authority and hub values are computed as follows:

$$Ap \leftarrow \sum_{q(q,p)\in G} Hq \qquad Hp \leftarrow \sum_{q(p,q)\in G} Aq$$

The authority value of page p is the sum of hub scores of all the pages that points to p, the hub value of page p is the sum of authority scores of all the pages that p points to(Fig.1).Iteration preceded on the neighborhood graph until the values converged.

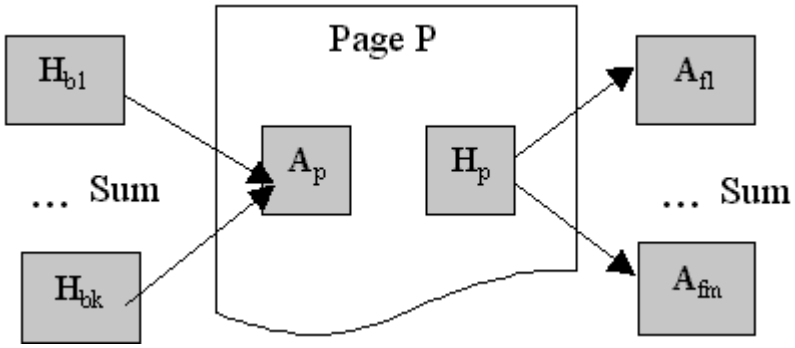

Fig.1 HITS

Kleinberg claimed that the less number of pages with the largest authority converged value should be the pages that had the best authorities for the topic. And the experimental results support the concept. Kleinberg also pointed out that there might be topic diffusion problems (the answer had a shift to a broader topic related to the query). And there might also be multi-communities for a query, where each community is focused on one meaning of the topic. Sometimes the first-principle community is too broad for the topic and the 2nd and 3rd community might contain the right answer to the users query.

5.2. PageRank Algorithm

The PageRank is the work of Brin and Page (1). It is used by Google search engine. Unlike HITS algorithm, PageRank is a query-independent algorithm, which is, assigning a rank score to each page independent of a given query; it is based on the connectivity

structure of the web pages. The score is assigned once and used for all subsequent queries.

The PageRank value of a page is weighted by each hyperlink to the page proportionally to the quality of the page containing the hyperlinks; i.e., the PageRank value of a page will spread evenly to all the pages it points to. As shown in Fig.2, page q_1 has two hyperlinks points to page p and page m separately, thus, the PageRank value of page q_1 will be distributed to p and m evenly, each of them gets 50 points. Since page p has two link points to it, the PageRank value of page p is the sum of weights of the incoming links.

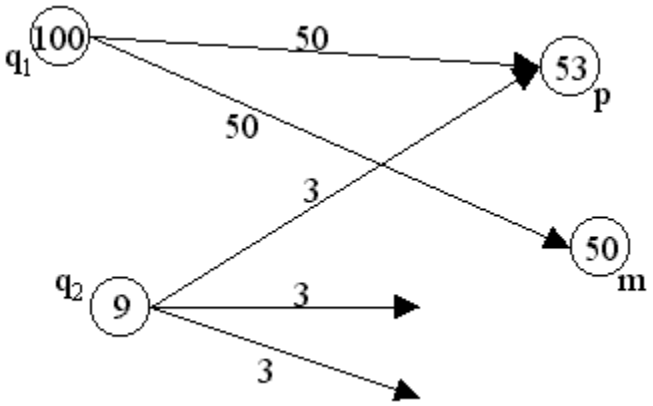

Fig. 2 PageRank

The PageRank R(p) of a page p can be defined as the probability that the surfer is at the page on a given time step:

$$PR(p) = (1-d) + d\ (PR(T_1)/C(T_1) + ... + PR(T_n)/C(T_n))$$

Where PR(p) is the PageRank of a page p
$PR(T_1)$ is the PageRank of a page T_1
$C(T_1)$ is the number of outgoing links from the page T_1
d is a damping factor in the range $0 < d < 1$, usually set to 0.85
The PageRank of a web page is therefore calculated as a sum of the PageRanks of all pages linking to it (its incoming links), divided by the number of links on each of those pages (its outgoing links).

5.3. STC Algorithm

This algorithm was developed by Zamir and Etzioni (7) in 1998. Based on this algorithm they have developed the clustering engine named Grouper. Suffix Tree Clustering (STC) is a linear time clustering algorithm that is based on identifying the phrases that are common to groups of documents. A phrase is an ordered sequence of one or more words. The base cluster is a set of documents that share a common phrase.

STC has three logical steps: (1) document "cleaning", (2) identifying base clusters using a suffix tree, and (3) combining these base clusters into clusters.

Step 1 - Document "Cleaning"

In this step, the string of text representing each document is transformed using a light stemming algorithm (delete word prefixes & suffixes and reducing plural to singular). Sentence boundaries (identified via punctuation and HTML tags) are marked and non-word tokens (such as numbers, HTML tags and most punctuation) are stripped.

Step 2 - Identifying Base Clusters

The identification of base clusters can be viewed as the creation of an inverted index of phrases for the document collection. This is done efficiently using a data structure called a suffix tree. This structure can be constructed in time linear with the size of the collection, and can be constructed incrementally as the documents are being read. The idea of using a suffix tree for document clustering was first introduced in 1997. Each node of the suffix tree represents a group of documents and a phrase that is common to all of them. Therefore, each node represents a base cluster. Furthermore, all possible base clusters (containing 2 or more documents) appear as nodes in our suffix tree. Each base cluster is assigned a score that is a function of the number of documents it contains, and the words that make up its phrase.

Step 3 - Combining Base Clusters

Documents may share more than one phrase. As a result, the document sets of distinct base clusters may overlap and may even be identical. To avoid the proliferation of nearly identical clusters, the third step of the algorithm merges base clusters with a high overlap in their document sets (phrases are not considered in this step The STC algorithm is incremental and order independent. As each document arrives from the Web, we "clean" it and add it to the suffix tree. Each node that is updated (or created) as a result of this is tagged. We then update the relevant base clusters and recalculate the similarity of these base clusters to the rest of the base clusters. if there is any changes in the base cluster graph result in any changes to the final clusters. The final clusters are scored and sorted based on the scores of their base clusters and their overlap. As the final number of clusters can vary, the top few clusters need to be reported. Typically, only the top 10 clusters are of interest. For each cluster reported, the number of documents it contains, and the phrases of its base clusters. In STC, as documents may share more than one phrase with other documents, each document might appear in a number of base clusters. Therefore a document can appear in more than one cluster. Note that the overlap between clusters cannot be too high, otherwise they would have been merged into a single cluster.

6. Some Experiments on Web page Clustering

Several clustering algorithms are developed in order to cluster the documents both in database management system and web page organization. Using different algorithms several clustering engines are developed, which cluster the web pages retrieved by the search engines automatically. Some are functioning as metasearch engine like metacrawler.com and vivisimo.com. Others are like Grouper and Retriever etc are the research test bed of the clustering techniques. The web page clustering is a new area of research where it applies statistics, computer science includes artificial intelligence. Followings are few examples of Web page clustering engine. It is important to note that the goal of clustering web pages is to effectively organize the retrieval information.

6.1 Experiment by HE et.al.

He et.al.(3) gathered three retrieval datasets, each corresponding to a one-word query: amazon, star, apple submitted to a search engine. The results show that their method effectively distinguishes different topics mixed together in a dataset. They tried with the keyword "amazon" to a search engine. "amazon" has at least three meanings. One is related to amazon.com, one of the largest on-line shopping websites. Another is the famous rain forest in South America. And the third is the name of ancient female warriors from Alecto, a female ruled monarchy. Total 2294 web pages have been retrieved with the keyword amazon from the search engine. Applying the clustering algorithm, they found about 8 clusters,

1st cluster gave 3 web pages of amazon.com, which is located at 3 countries.

2nd and 3rd clusters gave 4 & 5 web pages respectively, which gave information about female issues

4th cluster gave 5 web pages, which listed web page of large on-line auction company formed by Sothby and amazon.com

Clusters 5, 6, 7 are not really relevant to the query amazon. The 5,6 and 7 gave the information about movie star war includes some online shopping company selling goods relating to star war, and other shops.

From the clusters , they found that no cluster has a focused topic on Amazon rain forest. Checking the entire dataset, the 2294 URLs, they only got a couple of web pages that mention about the rain forest. They don't form a cluster with significant size.

As for the third meaning of amazon, although female warrior did not appear directly as a distinct topic in any cluster, some clusters focus on female issues, or even bi-sexual issue. By examining the content of these pages, we think that these issues are extension of the original meaning of amazon as female warriors. Especially, the cluster on the topic of bi-sexuality is beyond their expectation. This method can identify the topics which are unknown to us but it is valuable for the user. There are clusters with all authorities from the same websites, such as the clusters 6 and 7.

6.2 Vivisimo

Vivísimo (6) 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. It uses a specially developed heuristic algorithm to group - or cluster - textual documents. This algorithm is based on an old artificial intelligence idea: a good cluster - or document grouping - is one, which possesses a good, readable description. So, rather than form clusters and then figure out how to describe them, they only form well-described clusters in the first place (2).

Vivísimo is doing hierarchical, document clustering, conceptual, on-the-fly techniques

6.2.1 Document clustering

Document clustering is the automatic organization of documents into groups or clusters. "Document clustering" differs from other techniques (classification, taxonomy building, tagging, etc.) in that it is fully automated: further human intervention is not needed, although in many applications the Clustering Engine installation can benefit from specific domain expertise, when it is available. The biggest challenge for document clustering has been to quickly find meaningful groups that are concisely described.

6.2.2 Hierarchical clustering

Instead of producing a flat list of groups, Vivísimo's Clustering Engine organizes groups into a hierarchy or tree, using a well-known "Windows Explorer"-style interface. This interface can be used with no training since it is quite intuitive. Users can zoom in on items of interest while keeping visible an overview of all the topics.

6.2.3 Conceptual clustering

Conceptual clustering methods interleave the process of forming groups with the step of describing them, much like people might do by hand. So, if Vivísimo's document clustering tries to form a group but judges that the group cannot be described concisely, accurately, and distinctively, the group is rejected. In contrast, many other approaches rely mainly on mathematical optimization, in which description of the groups is relegated to the end after the groups are formed, which has never worked well.

6.2.4 On-the-fly clustering

Clustering is done just before the user sees the search results, on the fly. There is no need to prepare anything beforehand, much less pre-process the entire document collection from where the results came. For the IT integrator, this means that there is no complicated interdependence between Vivísimo's products and, say, a search engine. The interface consists of the plain text (titles and abstracts) of the search results as listed in XML or an html page

7. Conclusion

Clustering is not a brand-new technique. This technique has been used in the statistics for last five decades. The IR community has explored document clustering as an alternative method of organizing retrieval results, but clustering has yet to be deployed on most major search engines. Industry analysts predict that Google and other major search engines will need to make use of clustering technology to stay competitive.

8. References

1. Brin, S and Page, L. The anatomy of a large scale hypertexual web search engine. Proc. Of 7th WWW conference, 1998
2. CMU trio develops Internet search tool that sorts results in helpful clusters. *http://www.post-gazette.com/pg/03174/195311.stm*
3. He, X., and Ding, C.H.Q., (etc). Automatic topic identification using web page clustering. *IEEE ICDM*, 2001.
4. J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Proc. 9th Ann. ACM-SIAM Symp. Discrete Algorithms*, ACM Press, New York, 1998, pp.668-677.
5. Retriever: Improving Web Search Engine Results Using Clustering. *http://citeseer.nj.nec.com/cache/papers/cs/13555/http:zSzzSzwww.cs.umbc.eduzSz~ajoshizSzweb-minezSzretriever.pdf/retriever-improving-web-search.pdf*
6. Vivisimo.com. *http://www.vivisimo.com*
7. Zamir, Oren and Etzioni, Oren**.** Web document clustering: a feasibility demonstration. Annual ACM Conference on Research and Development in Information Retrieval Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval Melbourne, Australia P. 46 - 54. 1998