IJCSC International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp. 119-123

CONTEXT DISAMBIGUATION IN WEB SEARCH RESULTS USING CLUSTERING ALGORITHM

Jyoti Bhagwani and Kapil Hande

Department of Computer Science and Engineering, Nagpur, (India).
E-mail: chandwani1@rediffmail.com, Kapilhande@gmail.com

ABSTRACT

In preeminence web organization communication plays a significant task. Clusterization of web documents is a smart way for current web stuff. Arranging all documents into collection of similar area make things easier and shorten seek period. Different individual of the humanity are in search for speckled pattern on a periodic intervals. At every stride of existence decision makers are taking verdict after apprehensive scrutiny of fitting sequence. Hence excellence of decisions eventually depends upon excellence of information. With the coming on internet technology the population of online information seekers is increasing beyond any broaden of thoughts.

However information which is reclaimed from World Wide Web suffers from few disadvantages. One of the major disadvantage about which there is a worldwide concern that is information reclaimed is not the same as information alleged. We usually come across pages that are not of concern while searching the web. This is partially due to a word or words in seek out query having different circumstance, the user obviously expecting to locate pages related to the context of concern. This research paper attempts to consider the usefulness of Web page clustering algorithms to overcome the disadvantages of mismatching between information most wanted and information regained. This paper also proposes a method to disambiguate perspective in web exploration results.

Keywords: Disambiguate, Meta search, World Wide Web, Search Engines, Representation Retrieval, Context.

1. INTRODUCTION

The amount of the data in the Internet still grow gradually, what causes complicatedness in finding important information in the World Wide Web (WWW). Even though Internet hunts directory for large number of web pages they solve this problem only moderately. In the response to a query, user gets a ranked list of web pages that more or less match to the entered keywords. It takes a long time to check all the results to find relevant ones, because all the search results are mixed together without respect to their content. This situation was the main motivation to introduce systems that help users to organize search results into different categories. One example of such systems is a clustering metasearch engine. The idea of the meta-search engine, described in this article, is to gather search results from different sources (in our experiments we have used search engines: Google, MSN and Yahoo) rather than crawl the web pages and index them in the database. Then, search results are clustered, and our meta search engine produces groups of web pages that are similar to each other.

Expeditious and efficacious retrieval of relevant information from the World Wide Web has always been an issue of concern before the community of internet users. Today billions of pages of heterogeneous information are available on the servers of search engines. An obvious expectation from the information seekers is to get accurate information which satisfies their search criterion. Moreover an unfinished agenda of seamless retrieval of relevant information is getting compounded owing to dramatic rise in the internet traffic. The issue of concern is how to make desired information available to the information seekers with greater degree of result accuracy. In short clustering technique categorizes and organizes the search results into semantically meaningful clusters. It goes without saying that in order to enhance the efficiency and efficacy of information retrieval system we must minimize the number of disc accesses. Lesser the disc accesses are the greater is the efficiency and overall efficacy. To the delight of the community of internet users the actual performance of information retrieval process can be boost up by successful implementation of web page clustering method.

However quite often clustering is confused with classification. There is an obvious difference between the two. In classification the objects are defined to predefined class but in clustering classes are also to be defined. In clustering the objects having similar properties are housed in one class of objects and a single access to the disc makes entire class available to the information seeker. One possible solution to improve the quality of search a new approach, called web page clustering was formulated. Instead of showing the user a long list of results itself, it is split into groups related to sub-topics. If the user is shown these groups, possibly with some keyword type descriptions, they can then select one (or more) that fit their perceived interests. Clustering algorithms attempt to

group web pages together based on their similarities; thus pages relating to a certain topic will hopefully be placed in a single cluster. This can help users both in locating interesting web pages 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.

2. FACTORS RELATED TO PROBLEM

What is 'Context disambiguation' problem?

Context disambiguation identifying the sense of a word based on context. The word sense disambiguation community works on largely language-based techniques such as verifying the grammatical form of the word. Context disambiguation deals different contexts of the query including multiple referents, rather than the sense.

Differences from 'text categorization' and neural net-based approaches.

The text categorization community and neural nets are concerned with similarity-based document classification. However, the problem here is to classify documents based on the context of the query used; not on the similarity. As the query is to be used as a parameter for classifying sets of documents, such approaches do not easily adapt to the problem due to their static structure. Furthermore, they do not provide the flexibility, to incorporate the search query as a special parameter.

2.1 Common Context Ambiguities

Cases where there are different contexts for a query are abundant. The common ones include: Multiple referents, Place names; two places having the same name. Names of people: There are many famous people with first names Michael. Different events in the same place: 'Mumbai blasts' has many referents, two of them being the blasts of 1993 and also of 2003. Word sense ambiguities: These are much less important in the context of the web.

3. RELATED WORK

Researchers have applied the entire standard clustering methods [2, 8, 15] to web page clustering like Hierarchical (agglomerative and divisive), Partitioning (Probabilistic, *k*-medodis, *K*-means), and many more. Many algorithms build on the standard methods by using web or document specific are: Suffix Tree Clustering (STC) and Lingo [10, 11], Extended suffix Tree clustering (ESTC), Vivisimo [17, 9].

3.1 STC Algorithm

STC is a linear time clustering algorithm that is based on a suffix tree which efficiently identifies sets of documents that share common phrases. STC treats a document as a string, making use of proximity information between words. STC is incremental and $O(n)$ time algorithm. STC succinctly summarizes clusters contents for users. STC has three logical steps: (1) Document cleaning, (2) identifying base clusters using a Suffix tree, and (3) combining these base clusters into clusters.

3.2 Vivisimo Algorithm

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 [14]. Vivísimo is doing hierarchical, document clustering, conceptual, on the-fly techniques.

3.3 Lingo Algorithm

It is able to capture thematic threads in a search result, that is discover groups of related documents and describe the subject of these groups in a way meaningful to a human. This algorithm first preprocess the web pages, second step is for frequent phrase extraction, third step is for cluster label induction, then cluster content discovery is carried out, last step is for final cluster formation.

3.4 QDC Algorithm

This uses the user's query as part of a reliable measure. The new algorithm has five key innovations: Base Cluster in densification, Cluster merging, Cluster splitting, Selection of cluster, Cleaning of Cluster.

3.5 Crawl Engine

A web crawler (also known as a web spider, web robot, or-especially in the FOAF community—web scatter) is a program or automated script which browses the World Wide Web in a methodical, automated manner. Other less frequently used names for web crawlers are ants, automatic indexers, bots, and worms. This process is called web crawling or spidering. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a website, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for spam).

A web crawler is one type of boot, or software agent. In general, it starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. Web crawlers typically identify themselves to a web server by using the User-agent field of an HTTP request. Web

site administrators typically examine their web servers' log and use the user agent field to determine which crawlers have visited the web server and how often. The user agent field may include a URL where the Web site administrator may find out more information about the crawler. Spam bots and other malicious Web crawlers are unlikely to place identifying information in the user agent field, or they may mask their identity as a browser or other well-known crawler.

It is important for web crawlers to identify themselves so Web site administrators can contact the owner if needed. In some cases, crawlers may be accidentally trapped in a crawler trap or they may be overloading a web server with requests, and the owner needs to stop the crawler. Identification is also useful for administrators that are interested in knowing when they may expect their Web pages to be indexed by a particular search engine.

3.6 Clustering Software

Crawl engine accepts the name of the search engine and then the phrase on which clustering will be done. After getting the name of the search engine it automatically. Crawl engine will thus record all the web documents being shown by the search engine and they will save in the client computer. After collecting all the documents clustering will be perform on them before performing QDC on them they will be converted into numerical form visits the website and then put the query in the search box of the search engine. And the search engine searches the documents in available with it. The result obtains are irrelevant and contain man unnecessary data. The result page obtain in the search engine is crawl. The crawl examines each and every results displayed by the existing search engine, this result are stored in the database along with the link, meta data etc. Crawl engine will thus record all the web documents being shown by the search engine and they will save in the client computer. After collecting all the documents clustering will be perform on them before performing QDC on them they will be converted into numerical form by using Following Logic:

Our cluster adding numbers to each and every word in the document and store it in the array. SO that we can process the words easily, since number are easily processes than the string or words. Suppose that we 100 words in the document, some of them are repeated in the document. Know some will be repeated and some will be individual. Now assume that we have overall 30 words many of which are repeated in the document to make total words to 100: Now suppose that the first word is "Engineering" and it's repeated twice in the document. And then we have an array a [30] [1]. Know the word will be represented as a [1][0] = 2 and second word is engineering which is repeated *qo* times the it will be stored as a [2][0] = 10. Similarly we can have all the words and being assign a number and their frequency i.e. number of occurrence is recorded. After getting the numerical references of the entire document in an array we will get the word which is being allotted to the words in the phrases we have to search. Suppose we want to search "automobile engineering" in the document and its allotted number 3 and 2 in the document. Then will find frequency of words phrases in the document.

After finding frequency we will group the document according to the number of occurrences of the phrase words in the document. When the complete phrase is found in the document, the document is given highest priority and then the occurrences of single word will be considered. Thus the document will be grouped in to number *s* of category depending upon their in the sentences. If the word occurs in the noun place it will be given more important.

4. NEED FOR ANOTHER CLUSTERING ALGORITHM

These algorithms produce clustering of low quality and producing semantically meaningless clusters. Semantically meaningful clusters are often small, missing many relevant pages and contain irrelevant pages. The problem is that these algorithms only use textual properties and static's of pages from the result set. Other algorithms such as partitioning and hierarchical algorithm [15] use data similarity measures [2] to construct clusters. But these similarity-based methods are not effective to produce semantically meaningful clusters as these are directly applied to page data. One way of improving web page clustering algorithms is to make better use of the textual properties of web pages. The semantic relationships between words is very useful information; for example, synonyms, hyponyms, meronyms, etc. [4]. Word Net [4] is a lexical reference system and is one source of this information. However, the data in these systems is incomplete, particularly for commercial, technical, and popular culture word usage. An alternate source, although less accurate and less informative, is to use global document analysis and term co occurrence statistics to identify whether terms are related or unrelated. The number of pages in multi-term search result sets can approximate term co-occurrence statistics. Providing the most relevant pages earlier in the results can reduce the time users spend searching. [1] Most clustering algorithms order the pages in the clusters by their position in the search results. [3] Such an ordering fails to use the additional information about the user's search goal, provided by the user selecting the cluster, so the most relevant pages may not be shown first.

5. PROBLEM FORMALIZATION ALGORITHM

Although various algorithms are available in literature for clustering of web pages but they are not producing optimized set of clusters.

5.1 Hierarchical clustering

Instead of producing a flat list of groups, Vivisimo'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. Conceptual clustering methods interleave the process of forming groups with the step of describing them, much like people might do by hand. So, if Vivisimo's document clustering tries to form a group but judges that the group cannot be described concisely, accurately, 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.

5.2 Hierarchical Agglomerative Clustering

QDC uses a Hierarchical agglomerative clustering algorithm to identify the sub-cluster structure within each cluster. The algorithm uses a distance measure to build a dendrogram for each cluster starting from the base clusters in the cluster. Each cluster is split by cutting its dendrogram at an appropriate point—when the distance between the lowest pair of sub-clusters falls below a threshold. This threshold means that any groups of base clusters that are not tightly interconnected with each other will be split. Using a higher threshold will lower the split point and increase the splitting frequency.

5.3 Preprocessing of Web Pages

Initially preprocessing of web pages is carried out. Preprocessing involves removal of various page elements and words from the pages like HTML tags, punctuation and non informative text. Also set of stop words, common and uninformative words like "the", "it", and "on" are removed. Porter's suffix stripping algorithm is used to perform stemming. Using this algorithm we can transform the word to their root form. For example - Words "compute", "computing", and "computed" are stemmed to "compute", "dogs" becomes dog".

5.4 Base Cluster Identification

A base cluster is described by a single word and it consists of all the pages containing that word. This algorithm computes the query distance of each base cluster, the distance from the query using normalized Google distance. If the query distance is low, terms are unambiguous and more specific. While terms with a high query distance tend to be broad and often ambiguous.

5.5 Cluster Merging

Construction of larger clusters by merging clusters together. Each cluster (c) is constructed from a set of base clusters (base (c)), and a cluster is described by the word that describes the cluster's largest base cluster. However, the set of pages in a cluster is not necessarily all the pages in its base clusters. A page is only included in the cluster if it is present in enough of the base clusters in the cluster. This threshold should increase with the number of base clusters in the cluster, but should not increase steeply.

5.6 Cluster Splitting

Each cluster now contains at least all the base clusters that relate to one idea; this is assured as single-link clustering merges all related clusters. But single-link clustering, even with our improved similarity function, can produce clusters containing multiple ideas and irrelevant base clusters due to cluster chaining (drifting). Such clusters need to be split. It is easier to split such a compound cluster than to prevent its formation in the first place, because the splitting can take into account the final cluster, whereas the merging process cannot. It uses a hierarchical agglomerative clustering algorithm to identify the sub-cluster structure within each cluster. The algorithm uses a distance measure to build a dendrogram for each cluster starting from the base clusters in the cluster.

5.7 Cluster Selection

At this stage, we have a small set of coherent clusters. However, there will still be more clusters than can be presented to the user. Algorithm needs to select the best subset of the clusters to present to the user. Ideally, these clusters should be high quality clusters that cover all the pages in the original set with minimal overlap. It uses the ESTC cluster selection algorithm [6] with an improved heuristic, H(C), to select a set of clusters to show the user. The ESTC cluster selection algorithm uses the heuristic with a 3-step look-ahead hill-climbing search to select a set of clusters to present to the user.

6. CONCLUSION

Clustering technique has been used in the statistics for last five decades. The IR community has explored web page 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. In this paper, we have done analysis of current web page clustering algorithms like STC, Vivisimo and Lingo algorithm with their advantages and Disadvantages. This paper also presented new algorithm for to optimize disambiguates in web search results.

REFERENCES

[1] J. Back, and C. Oppenheim. "A Model of Cognitive Load Forir: Implications for user Relevance Feedback Interaction". *Information Research*, **6 (2)**, 2001.

[2] P. Berkhin. "Survey of Clustering Data Mining Techniques". *Technical Report*, *Accrue Software, San Jose*, CA, 2002.

[3] H. Chen, and S. Dumais. "Bringing Order to the Web: Automatically Categorizing Search Results". *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 145-152, 2000.

[4] R. Cilibrasi, and P.M.B. Vitanyi. "Automatic Meaning Discovery Using google.www.cwi.nl/paulv/papers/amdug.pdf, 2004.

[5] M.D. Cock, and C. Cornelis. "Fuzzy Rough Set Based Web Query Expansion". *International Workshop on Rough Sets and Soft Computing in Intelligent Agent and Web Technologies*, pp. 9-16, September, 2005.

[6] D. Crabtree, X. GAO, and P. Andreae. "Improving Web Clustering by Cluster Selection". *In the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 172-178, September, 2005.

[7] D. Crabtree, X. Gao, and P. Andreae. "Standardized Evaluation Method for Web Clustering Results". *In the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 280-283, September, 2005.

[8] A.K. Jain, M.N. Murty, and P.J. Flynn. "Data Clustering: A Review". *ACM Computing Surveys (CSUR)*, **31 (3)**, 264-323, 1999.

[9] F. Menczer. "Lexical and Semantic Clustering by Web Links". *Journal of the American Society for Information Science and Technology*, **55 (14)**, 1261-1269, December, 2004.

[10] S. Osinski, J. Stefanowski, and D. Weiss. "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition". *In Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference,Advances in Soft Computing*, pp. 359-368, Zakopane, Poland, 2004, Springer.

[11] S. Osinski, and D. Weiss. "A Concept-driven Algorithm for Clustering Search Results". *IEEE Intelligent Systems*, **20 (3)**, 48-54, May-June, 2005.

[12] M.F. Porter. "An Algorithm for Suffix Stripping". *Program*, **14 (3)**, 130-137, July 1980.

[13] A. Schenker, M. Last, H. Bunke, and A. Kandel. "A Comparison of Two Novel Algorithms for Clustering Web Documents". *In Proceedings of the 2nd International Workshop on Web Document Analysis (WDA 2003)*, pp. 71-74, Edinburgh, Scotland, August 2003.

[14] Vivisimo.com. http://www.vivisimo.com

[15] M. Steinbach, G. Karypis, and V. Kumar. "A Comparison of Document Clustering Techniques". *In KDD Workshop on Text Mining*, 2000.

[16] Daniel Crabtree, Peter Andreae, "Xiaoying Gao Query Directed Web Page Clustering". 2006 IEEE/WIC/ACM International Conference.

[17] Y. Wang, and M. Kitsuregawa. "On Combining Link and Contents Information for Web Page Clustering". *In 13th International Conference on Database and Expert Systems Applications*. DEXA2002, Aix-en-Provence, France, pages 902-913, September 2002.

[18] O. Zamir, and O. Etzioni. "Web Document Clustering: A Feasibility Demonstration". *In Research and Development in Information Retrieval*, pp. 46-54, 1998.