Cluster Based Web Search
Dipali Avinash Nikam
Dept of CSE, Dr. J.J. Magdum College of Engg. Jaysingpur (MS), d_a_kale@yahoo.co.in
Amol B. Rajmane
Dept of CSE, Dr. J.J. Magdum College of Engg. Jaysingpur (MS), amolbrajmane@yahoo.com

Abstract: The rapid growth of the Internet has made the Web a popular place for collecting information. Internet user access billions of web pages online using search engines. Information in the Web comes from many sources, including websites of companies, organizations, communications and personal homepages, etc. Effective representation of Web search results remains an open problem in the Information Retrieval community and also it put burden of collecting pages relevant to query on user. We develop an approach to organize search results into semantically meaningful groups of web pages and present these to the user as clusters, one for each meaning of the query. With each cluster, our approach provide a summary description that is representative of the real entity associated with that cluster (for the query “mouse”, the summary description may be a list of words such as “a small mammal and a popular pet”). The user can hone in on the cluster of interest to her and get all pages in that cluster.

Index Terms- Web Search result, Clustering.

I. INTRODUCTION

Web search is difficult because it is hard for users to construct queries that are both sufficiently descriptive and sufficiently discriminating to find just the web pages that are relevant to the user’s search goal. Queries are often ambiguous: words and phrases are frequently polysemantic and user search goals are often narrower in scope than the queries used to express them. This ambiguity leads to search result sets containing distinct page groups that meet different user search goals. Often users must refine their search by modifying the query to filter out the irrelevant results. Users must understand the result set to refine queries effectively; but this is time consuming, if the result set is unorganized. The main contribution of this paper is for assisting users to comprehend the result set. The approach identifies semantically meaningful groups of web pages and presents these to the user as clusters. The clusters provide an overview of the contents of the result set and when a cluster is selected the result set is refined to just the relevant pages in that cluster.

II. RELATED WORK

There has been a large body of work on disambiguation, entity resolution. A review of the main work is presented here, but the review is not exhaustive. In the Web People Search via Connection Analysis [1], it is proposed that if clusters contained errors (multiple people merged into same cluster) the advantages of cluster based approach are not obvious. For the disambiguating people that have same name a novel algorithm is developed. The algorithm is based on significant entities such as the name of other persons, organizations & locations on each web page forming relationships between the person associated with the web page & the entities extracted. The authors have compared traditional people search supported by current search engines with the clustered entity search built on top of disambiguation algorithm. Whereas in Disambiguation Algorithm for People Search on the Web [2], the authors have developed a disambiguation algorithm & then studied its impact on people search. The proposed algorithm extracts significant entities such as names, organizations & locations on each web page by using extraction techniques. Also, it extracts & parses HTML & web related data on each web page. The algorithm uses Entity-Relationship Graph where entities are interconnected via relationships. The algorithm analyses several types of information like attributes, interconnections that exist among entities in the ER Graph.

In the Towards Breaking the Quality Curse: A Web-Querying Approach to Web People Search [3], a Web People Search (WePS) approach is proposed. It is based on collecting co-occurrence information from the web. In order to make clustering decisions, a sky-line based classification technique is developed which classifies the collected co-occurrence information. This technique handles dominance that presents in data & adopt given clustering quality measure. In the Grouping Search Engine Returned citations for Person Name Query [4], goal is to group search engine return citations, such that citation in each group relate to same person. This is based on the three facets: attribute, link & page similarity. In this, the confidence matrix is constructed for all facts then a grouping algorithm is applied on final confidence matrix for all facets. The output is groups of the search-engine returned citations, such that the citations in each group relate to same person. In the Web Page Clustering using Heuristic Search in the Web Graph [5], the authors have proposed a framework which tackle various information retrieval and Web mining tasks. The proposed framework makes the heuristic search viable in the vast domain of the WWW and applicable to clustering of Web search results and to Web appearance disambiguation. In Adaptive Graphical Approach to Entity Resolution [6], the authors have presented a graphical approach for entity resolution. The overall idea behind this is to use relationships & to look at the direct and indirect (long) relationships that exist between specific pairs of entity representations in order to make a disambiguation decision. In
terms of the entity-relationship graph that means analyzing paths that exist between various pairs of nodes.

III. OVERVIEW OF THE APPROACH

In this section, we provide an overview of all necessary components for implementing the cluster based web search. We take middleware based approach for implementation. In the proposed approach, the processing of user query consists of the steps such as: A user submits a query to the middleware via a specialized Web-based interface. The middleware queries a search engine with this query via the search engine API and retrieves a fixed number (top K) of relevant web pages. The result is a set of clusters of these pages with the aim being to cluster web pages based on association to real entity. A set of keywords that represent the web pages within a cluster is computed for each cluster. The goal is that the user should be able to find the interest by looking at the sketch. The work has been divided into four modules which are as follows: Web pages retrieval for the query, Preprocessing of web pages, Clustering & its Processing, Graph Creation. The fig. 1 shows the proposed scheme. The following sections elaborate all of these steps in detail.

A. WEB PAGES RETRIEVAL

Web Pages retrieval for query can be implemented in many ways. There are many algorithms to process Top-k retrieval, for example: Fagins Threshold Algorithm (TA), No Random Access Algorithm (NRA) and Combined Algorithm (CA). All these threshold algorithms work on inverted indices for query terms. However taking into consideration the huge size of the web corpus, this process becomes very unfeasible. The HttpServlet component seeks to fill this void by providing an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.

B. PROCESSING OF WEB PAGES

After retrieving the top pages related to the query, the pages are processed by using IR techniques namely stemming & stop word removal. The job is to generate a set of highly relevant documents for any search query, using the available parameters on the web. The task is challenging because the available parameters usable by the algorithm are not necessarily the same as the ones web users see when deciding if a webpage is relevant to their search.

B.1 Stemming Algorithm

Stemming algorithms are used to transform the words in texts into their grammatical root form, and are mainly used to improve the Information Retrieval System’s efficiency. To stem a word is to reduce it to a more general form, possibly its root. If the input is connected/connecting/connection/connections then the output of the system is connect. Though the stem of a word might not be its root, we want all words that have the same stem to have the same root. The effect of stemming on searches of English document collections has been tested extensively. Several algorithms exist with different techniques. The most widely used is the Porter Stemming algorithm. In some contexts, stemmers such as the Porter stemmer improve precision/recall scores [7].

The stemmer operations are classified into rules where each of these rules deals with a specific suffix and having certain condition(s) to satisfy. A given word’s suffix is checked against each rule in a sequential manner until it matches one, and consequently the conditions in the rule are tested on the stem that may result in a suffix removal or modification. Using (VC)^m to denote VC repeated m times, this may again be written as [C](VC)^m[V].m will be called the measure of any word or word part when represented in this form. The case m = 0 covers the null word. The steps of algorithms were discussed in details in [7]. The algorithm is careful not to remove a suffix when the stem is too short, the length of the stem being given by its measure, m. It was merely observed that m could be used quite effectively to help decide whether or not it was wise to take off a suffix.

B.2 Stop Word Removal

After stemming it is necessary to remove unwanted words. There are 400 to 500 types of stop words such as “of”, “and”, “the,” etc., that provides no useful information about the document’s topic. Stop-word removal is the process of removing these words. Stop-words account for about 20% of all words in a typical document. These techniques greatly reduce the size of the search engine’s index. Stemming alone can reduce the size of an
index by nearly 40%. To compare a webpage with another webpage, all unnecessary content must be removed and the text put into an array. The Sam Allen has proposed the stop word dictionary on Dot Net Perls but this is not sufficient for some of the applications. For this reason, static dictionary is modified. If the input is “I saw a cat and a horse” then output is “saw cat horse” The fig 2 gives details of stop word removal.

1. Take the query from user.
2. Declare the dictionary of stop words.
3. Split parameter into words.
4. Allocate new dictionary to store found words.
5. Store results in this String Builder.
6. Loop through all words.
7. Convert to lowercase.
8. If this is a usable word, add it.
9. Return string with words removed.
10. Display query without stop words.

Fig 2: Algorithm for stop word removal

C. CLUSTERING AND ITS PROCESSING

When designing a Cluster Based Web Search, special attention must be paid to ensuring that both content and description (labels) of the resulting groups are meaningful to humans. As stated, “a good cluster—or document grouping—is one, which possesses a good, readable description.” There are various algorithms such as K-means, K-medoid but these algorithms require as input the number of clusters. A Correlation Clustering (CC) algorithm is employed which utilizes supervised learning. The key feature of Correlation Clustering (CC) algorithm is that it generates the number of clusters based on the labeling itself & not necessary to give it as input but it is best suitable when query is person names.

For general query, the algorithms are Query Directed Web Page Clustering (QDC), Suffix Tree Clustering (STC), Lingo, and Semantic Online Hierarchical Clustering (SHOC). The focus is made on Lingo because the QDC considers only the single words. The STC tends to remove longer high quality phrases, leaving only less informative & shorter ones. So, if a document does not include any of the extracted phrases it will not be included in results although it may still be relevant. To overcome the STC’s low quality phrases problem, in SHOC two novel concepts are introduced: complete phrases and a continuous cluster definition. The drawback of SHOC is that it provides vague threshold value which is used to describe the resulting cluster. Also, in many cases, it produces unintuitive continuous clusters.

The majority of open text clustering algorithms follows a scheme where cluster content discovery is performed first, and then, based on the content, the labels are determined. But very often intricate measures of similarity among documents do not correspond well with plain human understanding of what a cluster’s “glue” element has been. To avoid such problems Lingo reverses this process—first attempt is to ensure that we can create a human-perceivable cluster label and only then assign documents to it. Secondly, extract frequent phrases from the input documents, hoping they are the most informative source of human-readable topic descriptions. Next, by performing reduction of the original term-document matrix using Singular Value Decomposition (SVD), try to discover any existing latent structure of diverse topics in the search result. Finally, match group descriptions with the extracted topics and assign relevant documents to them. The detail description of Lingo algorithm is in [8]. The Fig 3 shows Lingo in the form of pseudo-code.

The system was assessed for a number of real-world queries; also analyzed the results obtained from our system with respect to certain characteristics of the input data. The queries are mainly categorized in four types such as ambiguous, general, compound and people name. The system was tested for all these queries & the result obtained is satisfactory. The fig. 4 shows the clusters obtained for the query “mouse”. Whereas the fig. 5 shows the relevant pages under the cluster “Computer Mouse.” From each type of query, some sample queries are taken for the testing of system. The Table 1 shows the name of query under each type, its clusters obtained. From the list of clusters obtained, only first 15 clusters are shown in the Table 1.

<table>
<thead>
<tr>
<th>Phase 1 Preprocessing</th>
</tr>
</thead>
<tbody>
<tr>
<td>for each document</td>
</tr>
<tr>
<td>{ Apply stemming;</td>
</tr>
</tbody>
</table>
| Mark stop words ;}
| Phase 2 Frequent Phrase Extractions |
| Discover frequent terms & phrases; |
| Phase 3 Cluster Label Induction |
| Use LSI to discover abstract concepts; |
| for each abstract concept |
| { find best-matching phrase; } |
| Prune similar cluster labels; |
| Phase 4: Cluster content discovery |
| for each cluster label |
| { use VSM to determine the cluster contents;} |
| Phase 5: Final cluster formation |
| Calculate cluster scores; |
| Apply cluster merging; |

Fig 3: Lingo-Main Phases Pseudo code
<table>
<thead>
<tr>
<th>Query Type</th>
<th>Query</th>
<th>Clusters obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambiguous</td>
<td>Mouse</td>
<td>Computer Mouse, Mickey Mouse, Website, Cells from Mouse, Gaming Mouse, Common Mouse, House Mouse, Technology, Gene, Graphics, Series, Apple, Magic Mouse, Support, Windows</td>
</tr>
<tr>
<td>General Terms</td>
<td>Yellow pages</td>
<td>Search Yellow pages, Local Business Directory, White Pages, Local Business Listings, Phone numbers, Maps &amp; Directions, TelephoneDirectory, Companies, Classifieds, International, People, Source, City Guides, Complete, Global</td>
</tr>
<tr>
<td>Compound Query</td>
<td>To be or not to be</td>
<td>Google Buchsuche-Ergebnisseite, Games, To be T, Face book, Name To be, News, Reviews, Seite, Service, People, Play Games, Question, Search, To be, Hooper, Years</td>
</tr>
</tbody>
</table>

**Table 1**: First 15 clusters for various queries

### D. GRAPH CREATION

It is a graphical approach, as it visualizes the dataset as the standard entity-relationship graph. There are other graphical disambiguation approaches, which visualize different graphs: Web Graph, Co-reference dependence graph, Entity-relationship graph (ER graph). Existing techniques are frequently based on probabilistic methodologies, application rely primarily on the mathematical apparatus from the area of Operation Research. The suitable visualization is the ER graph. By using JGraph class objects and their relations are displayed. A JGraph object doesn’t actually contain the data; it simply provides a view of the data. Like any non-trivial Swing component, the graph gets data by querying its data model. The summary line for above discussion is that this work is to help workers and researchers effectively sift through the large and often complex sets of search results.

### IV. CONCLUSION

Being a fairly young discipline of Computer Science, web search results clustering is gaining increasingly promising perspectives for the future. With this paper, we contribute to the scientific and the practical trend in web search clustering. The main result of this work is the design of a description-oriented web search results clustering. Following the analysis of present theory, we have decided to adopt an unconventional approach in which good cluster labels are identified first and then, based on them, the actual assignments of search results to groups are made.

To verify the practical value of idea, LINGO is implemented as a component of the framework. During the implementation work, additional factors were identified that
Significantly influence the quality of clustering. The key observation is that proper preprocessing of the input data is of crucial importance in web mining tasks. A similar observation has been made by other researches [9]. However, according to the analyses, the main goal of this paper has been attained – LINGO produces reasonably described and meaningful clusters. Nevertheless, there are still many problems which did not manage to address. The list of research directions are as:

Improvements in LINGO, Improvement in the preprocessing techniques because enormous influence of the preprocessing phase on the overall quality of clustering.

REFERENCES
8. Stanislaw Osinski, Jerzy Stefanowski, and Dawid Weiss “Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition”,

D.A.Nikam received the BE degree in Computer Science and Engineering from DKTE Society’s Textile and Engineering Institute, Ichalkaranji, India in 2004 and M.Tech in Computer Science and Technology from Shivaji University, Kolhapur, India in 2010. From 2004 to 2007, she worked as lecturer at DKTE Society’s Textile and Engineering Institute, Ichalkaranji. Since 2007, she is working as Assistant Professor at Dr.J.J.Magdum College of Engineering, Jaysingpur. She has published various papers in data mining and web search.

A.B.Rajmane received the BE degree in Computer Science and Engineering from Walchand College of Engineering, Sangli, India in 2000 and M.E in Computer Science and Engineering from Walchand College of Engineering, Sangli, India in 2007. Since 2002, he is working as Assistant Professor at Dr.J.J.Magdum College of Engineering, Jaysingpur. He has published various papers in data mining and web mining.