Performance Evaluation of Various Search Result Clustering Algorithms Survey

Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy E. Eassa
Faculty of Computing and Information Technology
King AbdulAziz University
Jeddah, Saudi Arabia

***Abstract*—Search Result clustering attempt to solve multiple meaning problem by automatic organizing a linear list of document references returned by a search engine into a set of meaningful thematic categories. Results Clustering dramatically reduce search time and effort, where search results clustering organize the search results into topics, fully automatically and without external knowledge. This paper proposes a survey on various Search result clustering engines and main three clustering search result algorithms and evaluate the quality these algorithms.**

Keywords— Srearch Result Clustering, STC Algorithm, Lingo Algorithm, TRSC Algorithm

I. INTRODUCTION:

Information on the Web is very huge in size. There is a need to use this big volume of information efficiently for effectively satisfying the information need of the user on the Web. Search engines become the major breakthrough on the web for retrieving the information. Where, among users looking for information on the Web, 85% submit information requests to various Internet search engines. Search engines are critically important to help users find relevant information on the Web.

Search engines in response to a user's query typically produces the list of documents ranked according to closest to the user's request. These documents are presented to the user for examination and evaluation. Web users have to go through the long list and inspect the titles, and snippets sequentially to recognize the required results. Filtering the search engines' results consumes the users' effort and time especially when multiple sub-topics of the given query are mixed together [1].

Search results clustering is an attempt to solve multiple meaning problems by automatic organizing a linear list of document references returned by a search engine into a set of meaningful thematic categories. Results Clustering dramatically reduce search time and effort, where search results clustering organize the search results into topics, fully automatically and without external knowledge. Designing a web search clustering algorithm is a big challenge because we have to ensure that both content and description (labels) of the resulting groups are meaningful to humans [2].

Most of the open text clustering algorithms follows a scheme where content clustering is performed based on the snippet, accordingly the labels are identified. This process does not focus on the cluster label where readable and unambiguous labels of the thematic groups are an important factor of the overall quality of clustering. This paper proposes a comprehensive survey on various search result clustering algorithms and their quality in the process of information retrieval from the Web.

The paper is organized as follows. Overview of search result clustering is introduced in Section 2. The goal of search result clustering is defined in Section 3. In Section 4, we describe main search result clustering algorithms. The performance evaluations are presented in Section and quality evaluations of the clustering search result algorithms are presented in Section 6. Finally we conclude the paper and give some future works in Section 7.

II. OVERVIEW OF SEARCH RESULTS CLUSTERING

While search engines are definitely good for certain search tasks such as finding the home page of an organization, they may be less effective for satisfying broad or ambiguous queries. Trying to solve this problem there are different methods in Web information retrieval systems are based on categorizing method. They categorize the whole Web, whether manually or automatically, and then letting the user see the results associated with the categories that best match his or her query.

However, main method is search results clustering, which consists of organizing the results into labeled list (also called categories). This method combines the best features of query-based and category-based search, in that the user may focus on a general topic by a weakly-specified query, and then drill down through the highly-specific themes that have been dynamically created from the query results. The main advantage of the cluster hierarchy is that it makes for shortcuts to the items that relate to the same meaning. In addition, it allows better topic understanding and favors systematic exploration of search results [3]; here we list the main clustering search engines:

- Grouper [4] is a document clustering interface to the HuskySearch meta-search service. HuskySearch retrieves results from several popular Web search engines, and Grouper clusters the results as they arrive using the Suffix Tree Clustering (STC) algorithm.

- Carrot2 [5] is an Open Source Search Results Clustering Engine. It can automatically organize small collections of documents (search results but not only) into thematic categories. Carrot2 combines several search results clustering algorithms: STC, Lingo, TRSC, clustering based on swarm intelligence (ant-colonies), and simple agglomerative Techniques.

- Vivísimo [6] is a major breakthrough, whose clusters and cluster labels were dynamically generated from the search results. Vivísimo 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 won the "best meta-search engine award" assigned by SearchEngineWatch.com from 2001 to 2003.

- WICE (Web information clustering engine) devise an algorithm called SHOC (semantic hierarchical online clustering) [7] that handles data locality and successfully deals with large alphabets.

- WebCAT [8] was built around an algorithm for clustering categorical data called transactional k-Means. Originally developed for databases, this algorithm has little to do with transactional processing and is rather about careful definition of dissimilarity (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.

- SnakeT [9] is both the name of the system and the underlying algorithm. An additional interesting feature of SnakeT is that it builds a hierarchy of possibly overlapping folders. It introduced novel features called approximate sentences—in essence non continuous phrases (phrases with possible gaps).

III. Web Clustering Engines Objectives

Traditional search engines are usually quite effective when the user has a particular URL to find or when the user is interested in some Web-mediated activity. But they can fail when the user has information need to satisfy, this is especially true for informational searches expressed by vague, broad or ambiguous queries.

A clustering engine attempt to solve the limitations of current search engines by providing categorized results as an additional feature to their standard user interface. In most clustering engines the categorized list are kept separated from the search result list that allowed users to use the two lists. The clustering engines can be most helpful in complementing the output of plain search engines are the following [3]:

- Fast retrieval: clustering search result facilitate the retrieval of more information on specific subtopics of interest, for example: if the user want to access the information about the same subtopic, s/he can use clustering engines because documents that related to the same subtopic are correctly gathered within the same cluster and the user is able to choose the right path from the cluster label.

- Multiple meaning: when queries are ambiguous, the user can view the cluster list which provides a high-level view of the whole query meaning.

In the next section we focus on search results clustering algorithms—the core component of a Web clustering engine—discussing how its features.

IV. Search Result Clustering Algorithms

There are several search results clustering algorithms, for example: STC, Lingo and TRSC. The algorithms differ in terms of the main clustering principle that leads to be different in their quality and performance characteristics. This section describes briefly these algorithms.

A. Suffix Tree Clustering Algorithm

Suffix tree (also called PAT tree) for a string is a data structure that presents the string as tree whose edges are labeled with strings, such that each suffix of string corresponds to exactly one path from the tree's root to a leaf [10]. Suffix tree way of a given string that allows for a particularly fast implementation of many important string operations. The Suffix Tree is a widely used data structure for efficient string matching.

Suffix Tree Clustering Suffix Tree Clustering (STC) algorithm were proposed by Zamir and Etzioni [11] used in their meta-search engine which is a linear time clustering algorithm that relies on identifying that common phrases in groups of documents. They treat a document as a string. All the suffixes of a document are identified, i.e., the first word to the last word, the second to the last, and so on. A suffix tree can be constructed by adding all these suffixes of the documents. The common phrases shared by some documents can then be identified. In other words, the documents can be clustered by the phrases they share. The beauty of this approach is it shows the user what are underlying the clusters. As a result, it is also very useful to interactive information retrieval [12].

B. Lingo Clustering Algorithm

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 [13].

To avoid such problems the Lingo algorithm was developed [14]. Lingo is particularly suited to solving the problem of search result clustering. Lingo reverses this process—it first attempts to ensure that we can create a human-perceivable cluster label and only then assign documents to it. Specifically, it extracts 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 SVD, it tries to discover any existing latent structure of diverse topics in the search result. Finally, it matches group descriptions with the extracted topics and assign relevant documents to them. This reversed process, compared to other search results clustering algorithms, allows Lingo to partially avoid the trap of verbally unexplainable clusters [13].

C. The Tolerance Rough Set Clustering algorithm

The Tolerance Rough Set Clustering (TRSC) algorithm is based primarily on the K-means algorithm presented in [15]. By adapting K-means clustering method, the algorithm remain relatively quick (which is essential for online results post processing) while still maintaining good clusters quality. The usage of Tolerance Space and upper approximation to enrich inter-document and document-cluster relation allows the algorithm to discover subtle similarities not detected otherwise. As it has been mentioned, in search results clustering, the proper labeling of cluster is as important as cluster contents quality. Since the use of phrases in cluster label has been proven [16] to be more effective than single words, TRSC algorithm utilize n-gram of words (phrases) retrieved from documents inside cluster as candidates for cluster description.

It is widely known that preprocessing text data before feeding it into clustering algorithm is essentials and can have great impact on algorithm performance. In TRSC, the following standard preprocessing steps are performed on snippets: text cleansing, text stemming, and Stop-words elimination. As TRSC utilizes Vector Space Model for creating document-term matrix representing documents, in document representation building step, two main standard procedures: index term selection and term weighting are performed [17].

V. Performance Evaluation

Lancaster and Fayen [18] listed six criteria for assessing the performance of information retrieval systems. They are: 1) Coverage, 2) Recall, 3) Precision, 4) Response time, 5) User effort, and 6) Form of output. Although the criteria were set up more than two decades ago and a great deal has been done to reduce user effort (e.g., design friendly user interface) in using the system, they still seem quite applicable to evaluating information retrieval systems today. The performance evaluation for the pervious algorithms are presented in [3], in this paper we focus on the quality evaluation.

VI. Qulaty Evaluation

We conduct the experiments as following. We use ten queries of three different types, all of them were proposed in [19], those are listed as bellow table.

Table 1: Sample of the queries and query types used in the evaluation

Type	Queries
Ambiguous queries	apple, NLP, Pluto
Entery names	dell, disney, world war 2
General terms	health, flower, music
Complex queries	clustering search results

Evaluating the precision and recall of the document clustering system is not a well defined task because there is no single ordering the results. There are many possible approaches to this evaluation. If the cluster were perfectly labeled, so that the user always chose the most relevant cluster, then it would be sufficient to evaluate the precision clusters and their labels. So we used the evaluation criteria in [20].Clustered results are judged by human subject with three different criteria:

(1) whether results agree with the majority of results in the same cluster

(2) whether results agree with the cluster's title;

(3) Whether a cluster's title is meaningful and agreeable with majority of inside results.

These three criteria are named respectively as: (1) Precision without cluster title, (2) Precision with cluster title, and (3) Precision of cluster's title.

Usually, users only spend time on the top k results. In this experiment, only the first one hundred results are retrieved and clustered. Following are average results we get from above ten queries:

Table 2: Shows result of the Precision without cluster title for each algorithm

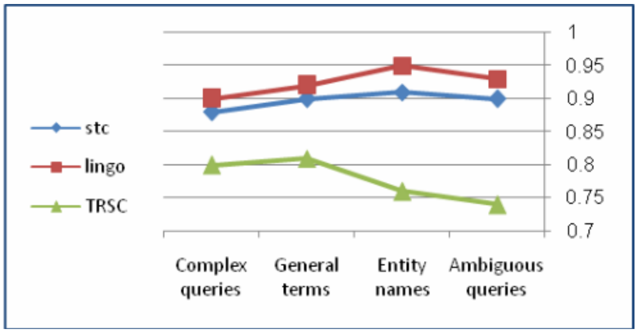

Fig 1: Precision of without cluster title for each algorithm

Table 3: Shows result of the Precision with cluster title for each algorithm

Query Types	**STC**	**Lingo**	**TRSC**
Ambiguous queries	0.86	0.87	0.8
Entity names	0.89	0.9	0.73
General terms	0.82	0.85	0.71
Complex queries	0.84	0.82	0.7
Average	0.8525	0.86	0.735

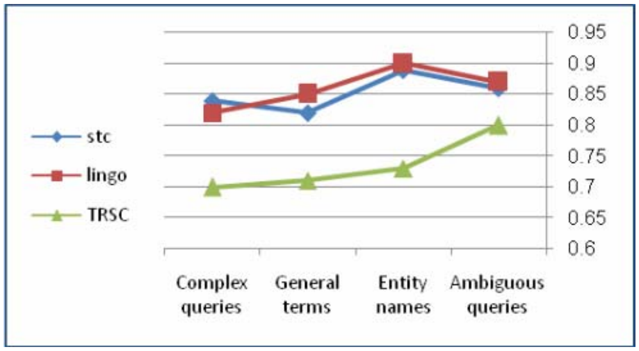

Fig 2: Precision of with cluster title for each algorithm

Table 4: Shows result of the Precision of cluster's title for each algorithm

Query Types	**STC**	**Lingo**	**TRSC**
Ambiguous queries	0.75	0.74	0.63

Entity names	0.72	0.7	0.63
General terms	0.73	0.76	0.61
Complex queries	0.71	0.73	0.59
Average	0.7275	0.7325	0.615

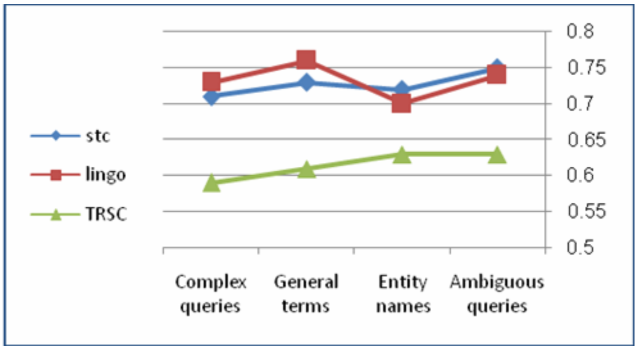

Fig 3: Precision of cluster's title for each algorithm

VII. Conclusion and Future Work

In this paper, we investigated the problem of how to cluster the search result from search engines. Queries are often ambiguous because many words have multiple meanings. By clustering the search results of the query term, it makes it easier for users to identify relevant results from the retrieved results. We review the main clustering engines and clustering algorithms. We evaluate the quality of the main three clustering search result algorithms: STC, Lingo and TRSC. We find TRSC is the worst and the other two algorithms have close results.

We plan to continue this research by adding semantic to the clustering search result algorithms and camper between the original algorithms and the modified algorithms.

References

[1] Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: "Learning to cluster Web search results. In: SIGIR '04". Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY,USA, ACM Press (2004) 210–217

[2] J. Lee, S. won Hwang, Z. Nie, and J.-R.Wen. Query result clustering for object-level search. In proc. SIGKDD, pages 1205–1214. ACM, 2009

[3] Carpineto, C., Osinski, S., Romano, G., &Weiss, D. (in press). A survey of Web clustering engines. To appear in ACM Computing Survey.

[4] Zamir O., Etzioni O. Grouper: A Dynamic Clustering Interface to Web Search Results. In Proceedings of the Eighth International World Wide Web Conference (WWW8), Toronto, Canada, May 1999

[5] Osiński, S. and Weiss, D. Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework. Springer Lecture Notes in Computer Science, vol. 3528, pp. 439—444, Proceedings of the third International Atlantic Web Intelligence Conference (AWIC 2005), Łodź, Poland

[6] S. Koshman, A. Spink, and B. J. Jansen, "Web searching on the vivisimo search engine," JASIST, vol. 57, no. 14, pp. 1875–1887, 2006.

[7] Zhang,D.And Dong,Y. 2004. Semantic, hierarchical, online clustering ofWeb search results. In Proceedings of 6th Asia-PacificWeb Conference (APWeb). Lecture Notes in Computer Science, vol. 3007. Springer, 69–78.

[8] F. Giannotti, M. Nanni, and D. Pedreschi. Webcat: Automatic categorization of web search results. In SEBD03

[9] Ferragina, P. And Gulli, A. 2004. The Anatomy of SnakeT: A Hierarchical Clustering Engine for WebPage Snippets. In Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases. Lecture Notes in Computer Science, vol. 3202. Springer, 506–508.

[10] "Suffix tree", in Wikipedia: The Free Encyclopedia; Wikimedia Foundation Inc; available from http://en.wikipedia.org/wiki/Suffix_tree; Internet; retrieved 4 February 2012.

[11] Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: SIGIR 1998, pp. 46-54 (1998)

[12] Cao, G., Song, D., and Bruza, P. Suffix tree clustering on post-retrieval documents, July 2003.

[13] Gauri Suresh Bhagat, Mrunal S. Bewoor, Suhas Pati , "IMPROVED SEARCH ENGINE USING CLUSTER ONTOLOGY ", International Journal of Advances in Engineering & Technology, Nov 2011, ISSN: 2231-1963

[14] Stanisław Osiński, Dawid Weiss. 2005. "A concept-driven algorithm for clustering search results". IEEE Intell. Syst. 20, 3, 48–54.

[15] Ho, T.B, Nguyen, N.B.: Nonhierarchical document clustering based on a tolerance rough set model. International Journal of Intelligent Systems 17 (2002) 199–21

[16] Osinski, S. An algorithm for clustering of web search result. Master's thesis, Poznan University of Technology, Poland, June 2003

[17] Chi Lang Ngo, Hung Son Nguyen: Tolerance rough set approach to clustering web search results, J.-F.Boulicaut, F.Esposito, F.Gianotti, and D.Pedreschi (Eds.): Knowledge Discovery in Databases: Proceedings of PKDD 2004, LNAI 3302, 513–517.

[18] Lancaster, F.W. and Fayen, E.G. "Information Retrieval On-Line." Los Angeles: Melville Publishing Co., 1973. Chapter 6.

Query Types	**STC**	**Lingo**	**TRSC**
Ambiguous queries	0.9	0.93	0.74
Entity names	0.91	0.95	0.76
General terms	0.9	0.92	0.81
Complex queries	0.88	0.9	0.8
Average	0.8975	0.925	0.7775

[19] Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy A. Essa, "Semantic Clustering Approach Based Multi-agent System for Information Retrieval on Web",IJCSNS International Journal of Computer Science and Network Security,Vol. 12 No. 1 pp. 41-46

[20] Khac Le, H. (2012). Inductive clustering: A technique for clustering search results . Retrieved from http://sifaka.cs.uiuc.edu/course/598cxz05s/report-hle.pd