ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Efficient Clustering Multiple Web Search Engine Results and Ranking

R. Rubini [1], Dr. R. Manicka Chezian [2]

Research Scholar, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, India[1]

Associate Professor, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, India[2]

ABSTRACT: World Wide Web is considered the most valuable place for Information Retrieval and Knowledge Discovery. Web search engines with effective and efficient techniques for Web service retrieval and selection becomes an important issue. Existing web search result based on keyword matching in single search engine only. This paper details a modular, self-contained web search results clustering system that enhances search results by (i) performing clustering on lists of web results returned by queries to search engines, and (ii) ranking the results and labeling the resulting clusters by using a calculated relevance value as a degree of membership to clusters. An efficient page ranking method is also proposed that orders the results according to both the relevancy and the importance of documents. Web search result clustering has been emerged as a method which overcomes these drawbacks of conventional information retrieval (IR) systems. This paper gives a sufficient overview and categorizes various techniques that have been used in clustering of web search results.

KEYWORDS: Clustering**,** Information Retrieval (IR), Text Mining, Web Mining, Web Search Engine**.**

I.INTRODUCTION

The process of retrieval is highly affected by the vague query put up by the average user. A way of assisting users in finding what they are looking for quickly is to group the search results by topic. There are many web clustering engines available on the web (Carrot2, Vivisimo, SnakeT, Grouper etc) which give the search results in forms of clusters. This process is usually seen as complementary rather than alternative and different to the search engine [1]. The main use for web search result clustering is not to improve the actual ranking, but to give the user a quick overview of the results. The Scatter/Gather system by is held as the predecessor and conceptual father of all web search result clustering. However, its current status is far from satisfaction for several possible reasons [2], such as different users have different requirements and expectations for search results; sometimes queries cannot be expressed clearly just in several keywords; Synonymous and polysemous words make searching more complicated etc.

II.RELATED WORKS

Oren Etzioni was the person who coined the term Web Mining first time [1]. Initially two different approaches were taken for defining Web Mining. First was a “process-centric view”, which defined as a sequence of different processes. Whereas, second was a “data centric view” defines a type of data [3].Web mining is also a cross point of database, information retrieval and artificial intelligence. The most common way of representing text documents is using the Vector Space Model (VSM) [12]. Each vector component has an associated weight which indicates the importance of that attribute to characterize or represent the document [4].

Oren Zamir and Oren Etzioni[1] in their research listed the key requirements of web document clustering methods as relevance, brows able summaries, overlap, snippet tolerance. They have given STC (Suffix Tree Clustering) algorithm which creates clusters based on phrase shared between documents. Most document clustering methods perform several pre-processing steps including stop words removal and stemming on the document set [3, 4]. Scatter/gather described in [5] was an early cluster based document browsing method that performed post retrieval clustering on top-ranked documents returned from a traditional information retrieval system.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

A. Limitations of Web Search

The internal relationships among the documents in the search result are rarely presented and are left for the user. Standard information retrieval systems rely on two orthogonal paradigms: the textual similarity with the query (e.g., tf-idf-based cosine similarity) on one hand and a query independent measure of each web page's importance (e.g., link authority ranking). The most famous examples of such ambiguous queries include bass (fish or instrument), java (programming language, island or coffee), jaguar (animal, car or Apple software) and IR application (Infrared application or Information Retrieval application).

III.PROPOSED APPROACH

Clustering of web search results has been studied in the area of Information Retrieval (IR). The goal of clustering search result is to give user an idea of what the result contains.. Document snippet clustering can be classified as the content-based clustering. Graph based clustering can be categorized as topology-based clustering.

A. Multiple Search engine Design

The most known general search engines are Google and Yahoo!, but one of the oldest search engines is AltaVista. All existing search engines have weaknesses, even Google. This part represents a real reason for building more search engine. Stream-based access allows all or a significant subset of pages to be retrieved as a stream [6]. Query-based access to the pages and the computed features (from the feature repository) is provided via the Web Base query engine. In selection of search engines twenty five search engines were selected to conduct our experiment. They are All the Web, AltaVista, google, yahoo, clusty, you tube, file tube, citeceer etc., to name a few. At first, the search engines were selected and the user query is submitted to all search engines under consideration. The queries covered a broad range of topics. The topics are as follows: Internet, literature, music, plants, sports, travel etc. The content of these pages is compared to give the result.

1. Web Crawler

This process continues until all reachable content has been gathered, until the refresh interval (refresh setting) is complete or until another configuration parameter limiting the scope of the crawl is reached. There are many different ways to adjust the configuration to suit a specific Web crawling scenario.

a. Controller Module - This module focuses on the Graphical User Interface (GUI) designed for the web crawler and is responsible for controlling the operations of the crawler. It controls the Fetcher and Parser.
b. Fetcher Module - This module starts by fetching the page according to the start URL specified by the user.
c. Parser Module - This module parses the URL's fetched by the Fetcher module and saves the contents of those pages to the disk. After that indexer create index in the database to organize the data by categorize them. The indexer extracts all the information from each and every document and stores it in a database. All high-quality search engines index each and every word in the documents and give a unique word Id. Then the word occurrences, which some search engines call "hits," are checked, recording all the words.

B. Web Result Filtering

Bloom Filter: A Bloom filter of a set U is implemented as an array of m bits. Each element u (u ∈ U) of the set is hashed using k independent hash function h1... . hk. Hash function hi(u) for $1 \leq i \leq k$ maps to one bit in the array {1 . . .m}. When an element is added to the set, it sets k bits, each bit is corresponding to a hash function, in the Bloom filter. If a bit was already set it stays 1[10]. For membership checks, Bloom filters may yield a false positive; it may appear that an element v is in U even though it is not. From the analysis, given n = |U| and the Bloom filter size m, the optimal value of k minimizes the false positive probability, pk, where p denotes probability that a given bit is set in the Bloom filter, is k = m n ln 2. Previously, Bloom filters have primarily been used for finding set-membership.

For finding similar documents, we compare the Bloom filter of one with that of the other. In case the two documents share a large number of 1's (bit-wise AND) they are marked as similar. In this case, the bit-wise AND can also be perceived as the dot product of the two bit vectors. If the set bits in the Bloom filter of a document are a complete subset of that of another filter then it is highly probable that the document is included in the other. Web pages are of

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

fragments, either static ones (e.g., logo images), or dynamic (e.g., personalized product promotions, local weather). When targeting pages for a similarity based "grouping", the test for similarity should be on the fragment of interest and not the entire page.

C. Cluster Web Result

Probability K-Means clustering is one of the most common clustering algorithms. Once the number of final clusters is decided, it needs to pick up K data points from data collection as the initial centroids for the first assignment of data points [7]. The assignment of all data points to different clusters is performed iteratively until some stop condition is reached. The principles of K-Means:

1. Pre-determine the K number of final clusters and randomly select the K data points as initial cluster centroids.
2. Assign each data point to the cluster that is closest to.
3. Re-compute K centroids after all data points have been assigned to corresponding clusters.
4. Repeat the step 2 and 3 until the some stop condition is reached.

Distance measure is usually the most common similarity metrics probability K-Means clustering uses, such as Squared Euclidean distance measure as shown in the Equation 1, where x1, x2, ..., xn is the representation of point X and y1, y2, ..., yn is the representation of point Y. (both Euclidean distance and Squared Euclidean distance don't consider the normalization) therefore, K-Means clustering uses cosine similarity metrics that is described previously in the section of "Vector Space Model"[11].

$$d = \sum_{i=1}^{n} (x_i - y_i)^2$$

Clustering system usually consists of documents crawling, indexing and clustering as its basic procedures. Our probability K-Means clustering method is implemented on top of Apache Lucene indexing, Apache Mahout Vector creation and K-Means clustering components. Several other tools such as Html Cleaner, Html Parser are used for parsing web page to get content fragments and out-links, and Yahoo! site explorer is used to retrieve in-links of certain page.

D.Click Based Ranking

The primary benefit of a click-through algorithm for web page ranking is that it incorporates actual user click-through behaviour to rank web pages which contrasts to the link-analysis algorithms, such as Page Rank, which do not build their rankings off of any actual usage but instead off of the underlying linked structure of the network of web pages on the Internet. The click-through model we propose herein specifically incorporates click-through from the search engine that is incorporating the rankings. The primary users of web page ranking systems are search engines [8]. Thus, we bypass the somewhat indirect logic of link analysis and the reputation system it is based on, which values pages based upon an apparent reputation in the world of web pages. Here we get directly to the point, attempting to answer the question, what pages do the *people* that visit search engine value? Once we have answered this question, it is interesting to go back and validate, critique, or complement the link analysis rankings with the results.

Assume the pages are ordered by the search engine in order of their indices: 1,2,3,... Then the following two values represent the expected probability that users eventually click-through and the expected number of pages views per user until a click-through.

E[probability of success] = $p1 + (1 ¡ p1)[p2 + (1 ¡ p2)[p3 + \ldots$ E[search time] = $1 + (1 ¡ p1)[1 + (1 ¡ p2)[1 + \ldots$

It is reasonable to assume that maximizing the first of these values and minimizing the second are both primary objectives for a search engine. Now, obviously with respect to our model and assumption 5) above, the probability of a click-through will be the same every time. However, if we relax assumption 5).In any case, if we can simply identify the *pi* values then we can optimize with respect to both of these objectives by simply ordering the *pi* values in decreasing order [9]. Notice, the probability of a click-through in *m* steps can be rewritten E[probability of success] = $1 ¡ (1 ¡ p1)(1 ¡ p2):::(1 ¡ pm)$ This value is decreasing in *pi*, so we want highest *pi*'s included for all sets of *m* steps.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Thus, we want *pi* ranked in decreasing order to maximize this probability for all *m*. Expected number of page examinations can also be rewritten as *E*[search time] = 1 + (1 ¡ *p*1) + (1 ¡ *p*1)(1 ¡ *p*2) + *::: +* (1 ¡ *p*1)*:::*(1 ¡ *pk*) so for any ordering of the pages, if we swap pages *i* and *j* where *i* was originally placed before *j*, the only terms in this sum that change are those that include a *pi* term and no *pj* term. These terms all decrease if *pj* > *pi* and increase if *pj* < *pi*. Thus, to minimize expected number of page examinations, we must order by largest *pi*.
We begin by modeling the system. We assume:
1. There are *k* web pages.
2. The search engine cannot distinguish between pages by topic. Every query of user is equally relevant to all pages.
3. Each page *i* has an inherent value parameter *pi* which represents the probability that any given user, upon examining page *i*'s listing on the search engine will click-through to page *i*.
4. The search engine produces for each user an ordered list of pages. Users examine these pages in order until they decide to click-through to a page..
5. Users will continue examining pages until they have either clicked-through to a page, or rejected all pages.

IV.EXPRIMENTAL RESULTS

These strategies are ranked with click based ranking algorithm as well as with a click-count ranking approach. This algorithm mainly deals with the concept of when the submitted query does not give the expected result then the links returned by the given query gives out the best result .Experimental results showed a better result by using this proposed algorithm against Click-Count and cluster results.

A. Performance Comparisons

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where,
- ✓ True positives (TP) - number of reviews correctly labeled as belonging to particular class (positive/negative).
- ✓ False positives (FP) - number of reviews incorrectly labeled as belonging to particular class.
- ✓ False negatives (FN) - number of reviews were not labeled as belonging to the particular class but should have been labeled.
- ✓

Table 1: Number of Query Vs Precision

Algorithms	***5***	***10***	***15***	***20***	***25***
TRECVID	**0.38**	**0.28**	**0.21**	**0.18**	**0.15**
Cluster with Ranking	**0.49**	**0.41**	**0.37**	**0.27**	**0.19**

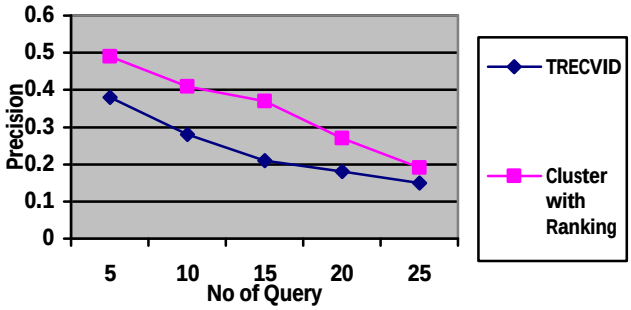

Fig. 1 Number of Query Vs Precision

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Table 2: Recall Vs Precision

Algorithms	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9	1
TRECVID	0.81	0.74	0.65	0.6	0.53	0.47	0.3	0.21	0.1
Cluster with Ranking	0.95	0.81	0.74	0.68	0.59	0.51	0.43	0.33	0.2

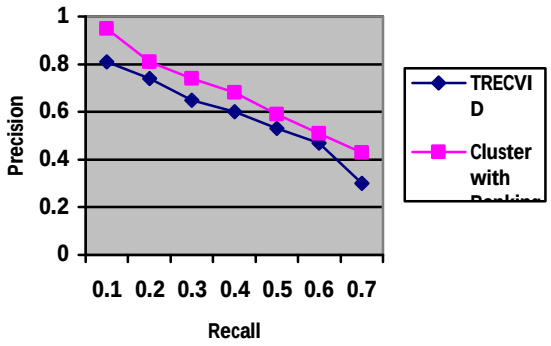

Fig.2 Comparison of Averaged Precision-Recall (PR)

Table 3: Different Search engine with Result

Web Search Engine URL	User Query	Normal Query Result	Cluster Based web result
Yahoo	Java	50	80
wikipedia	Java	30	60
isohunt	Java	10	40
torrenz	java	70	80

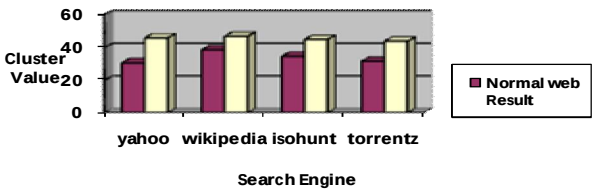

Fig.3 Different search engine with Cluster

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Table 4: Different Query with Cluster Size

Topic	Number of final clusters	Singleton cluster	Maximum Cluster Size	Number of clusters withsize >3
Data mining	15	66	13	5
Data mining	14	60	20	4
Data mining	12	53	20	8
Java	21	129	89	6
Java	23	105	107	8

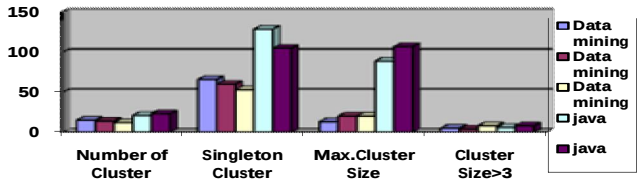

Fig.4 Different search engine with query

V.CONCLUSION AND FUTURE WORK

The important role of search engines in the World Wide Web, here improve the crawling process employed by multiple search engines with the goal of improving the quality of the service they provide to clients. Our analysis of the cluster, the web result and ranking as done, and the metric of embarrassment, which is introduced by a preferable goal. The next-generation Web architecture represented by the Semantic Web will provide adequate instruments for improving search strategies and enhance the probability of seeing the user query satisfied without requiring tiresome manual refinement. Future enhancement of Particle Swarm Optimization method based upon the concept of Swarm Intelligence is being implemented in high-dimensional sequence clustering analysis for web usage mining.

REFERENCES

1. O. Zamir and O. Etzioni, “Web Document Clustering: A Feasibility Demonstration,” Research and Development in Information Retrieval, pp. 46-54, 1998.
2. Hsinchun Chen and Michael Chau, “Web Mining: Machine learning for Web Applications”, Annual Review of Information Science and Technology 2003.
3. Schenker, M. Last and A. Kandel (2001),” A term-based algorithm for hierarchical clustering of Web documents” in Proceedings of the Joint 9th Canada IFSA World Congress and 20th NAFIPS International Conference, vol.5, pp. 3076-3081, Vancouver,, July 2001.
4. Nicholas O. Andrews and Edward A. Fox, “Recent Development in Document Clustering Techniques”, Dept of Computer Science, Virgina Tech 2007.
5. Ramakrishna, M.T. Gowdar, L.K. Havanur, M.S. Swamy (2010), “Web Mining: Key Accomplishments, Applications and Future Directions”, International Conference on Data Storage and Data Engineering (DSDE), pp.187 – 191, 2010.
6. Pawan Lingras ,Rui Yan and Chad West, “ Fuzzy C-Means Clustering of Web Users for Educational Sites”, Springer Publication ,2003.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

7. Guandong Xu," Web Mining Techniques for Recommendation and Personalization", Ph.D.dissertation, Victoria University, Australia, March 2008.
8. Andreas Hotho and Gerd Stumme, "Mining the World Wide Web- Methods, Application and Perceptivities", in Künstliche Intelligenz, July 2007.
9. WangBin and LiuZhijing , "Web Mining Research" , in Proceeding of the 5th International Conference on Computational Intelligence and Multimedia Applications(ICCIMA'03) 2003.
10. D.R. Cutting, D.R. Karger, J.O. Pederson and J.W. Tukey (1992). "Scatter/gather: a cluster-based approach to browsing large document collections", in Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'92, pp. 318-329, Copenhagen, Denmark, June 1992.
11. Salton, G., Wong, A., Yang, C.S. (1975). "A vector space model for automatic indexing". Communications of the ACM, 18(11):613-620, 1975.

BIOGRAPHY

R.Rubini is a research scholar in Nallamuthu Gounder Mahalingam College, Pollachi. She received her Master of Computer Application (MCA) in 2013. She has presented papers in International/National Conferences and attended Workshop, Seminar. Her research interest focuses on Data Mining.

Dr. R.Manickachezian received hisM.Sc., degree in Applied Science fromP.S.G College of Technology, Coimbatore, India in 1987. He completed his M.S. degree in SoftwareSystems from Birla Institute ofTechnology and Science, Pilani, Rajasthan, India and Ph Ddegree in Computer Science from School of Computer Science and Engineering, Bharathiar University, Coimbatore, India. He served as a Faculty of Maths and Computer Applications at P.S.G College of Technology, Coimbatore from 1987 to 1989. Presently, he has been working as an Associate Professor of Computer Science in N G M College (Autonomous), Pollachi under Bharathiar University, Coimbatore, India since 1989. He has published thirty papers in international/national journal and conferences: He is a recipient of many awards like Desha Mithra Award and Best Paper Award. His research focuses on Network Databases, Data Mining, Distributed Computing, Data Compression, Mobile Computing, Real Time Systems and Bio-Informatics.