# Implementation of Web Search Result Clustering System

Hanumanthappa M. and B.R. Prakash

Bangalore University, Bangalore
hanu6572@hotmail.com,
brp.tmk@gmail.com

**Abstract.** Web search results clustering is an increasingly popular technique for providing useful grouping of web search results. This paper introduces a prototype web search results clustering engine that use the random sampling technique with medoids instead of centroids to improve clustering quality, Cluster labeling is achieved by combining intra-cluster and inter-cluster term extraction based on a variant of the information gain measure by using Modified Furthest Point First algorithm. M-FPF is compared against two other established web document clustering algorithms: Suffix Tree Clustering (STC) and Lingo, which are provided by the free open source Carrot2 Document Clustering Workbench. We measure cluster quality by considering precision , recall and relevance. Results from testing on different datasets show a considerable clustering quality.

## 1 Introduction

With the increase in information on the World Wide Web it has become difficult to find the desired information on search engines. 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. It takes lot of time to find the relevant information. Typical queries retrieve hundreds of documents, most of which have no relation with what the user is looking for. The reason for this is due to the user failing to formulate a suitable or specific enough query, and efforts have been made to use some form of natural language processing when processing a search query to try and understand the underlying concept the user is trying to get across. One solution to this problem is to enable more efficient navigation of search results by clustering similar documents together [1][2]. By clustering web search results generated by a conventional search engine, the search results can be organized in a manner to reduce user stress and make searching more efficient while leveraging the existing search capability and indexes of existing search engines [9].

## 2 Overview of M-FPF and Improving the FPF Algorithm

In this paper we improve the Furthest Point First algorithm from both the computational cost point of view and the output clustering quality. Since theoretically the FPF

Aswatha Kumar M. et al. (Eds.): Proceedings of ICAdC, AISC 174, pp. 795–800.
springerlink.com

algorithm as proposed by Gonzalez [12] is optimal (unless P = NP), only heuristics can be used to obtain better results and, in the worst case, it is not possible to go behind the theoretical bounds. We profiled FPF and analyzed the most computational expensive parts of the algorithm. We found that most of the distance computations are devoted to find the next furthest point. FPF clustering quality can be improved modifying part of the clustering schema. We describe an approach that use the random sampling technique to improve clustering output quality, we call this algorithm M-FPF[10][11]. Another crucial shortcoming of FPF is that it selects a set of centers not representative of the clusters. This phenomenon must be imputed to the fact that, when FPF creates a new center, it selects the furthest point from the previous selected centers and thus the new center can likely be close to a boundary of the subspace containing the data set. To overcame this we modify M-FPF to use medoids instead of centers.

## 3 Overview of Clustering and Labeling System

(1) **Querying one or more search engines:** The query entered by the user is redirected to the selected search engines. As a result of the search engine, a list of snippets describing Web pages relevant to the query. An important system design issue is deciding the type and number of snippet sources to be used as auxiliary search engines.

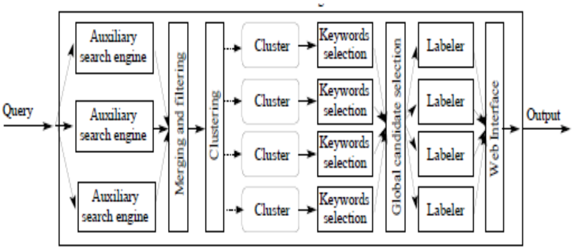

**Fig. 1.** Architecture of clustering and labeling system

With the rank of the snippet in the list returned by the search engine. Therefore the need of avoiding low-quality snippets suggests the use of many sources each supplying a low number of high-quality snippets.

(2) **Cleaning and filtering:** The input is then filtered by removing non-alphabetic symbols, digits, HTML tags, stop words, and the query terms. These latter are removed since they are likely to be present in every snippet, and thus are going to be useless for the purpose of discriminating different contexts. We then identify the language of each snippet, which allows us to choose the appropriate stop word list and stemming algorithm. Currently we use the ccTLD (Country Code Top Level Domain) of the url to decide on the prevalent language of a snippet.

**(3) First-level clustering:** We build a flat $k$-clustering representing the first level of the cluster hierarchy, using the M-FPF algorithm and the Generalized Jaccard Distance [5]. An important issue is deciding the number $k$of clusters to create. Currently, by default this number is fixed to 30, but it is clear that the number of clusters should depend on the query and on the number of snippets found. Therefore, besides providing a default value, we allow the user to increase or decrease the value of $k$to his/her liking. Clusters that contain one snippet only are probably outliers of some sort, and we thus merge them under a single cluster labeled “Other topics”.

**(4) Snippets re-ranking:** In general users are greatly facilitated if the snippets of a cluster are listed in order of their estimated importance for the user. Our strategy is to identify an “inner core” of each cluster and “outliers”. In order to achieve this aim we apply the FPF algorithm within each cluster as follows. Since FPF is incremental in the parameter $k$, we increment $k$up to a value for which it happens that the largest obtained cluster has less than half of the points of the input cluster.

**(5) Candidate words selection:** For each cluster we need to determine a set of candidate words for appearing in its label called as candidates. For each word that occurs in the cluster we sum the weights of all its occurrences in the cluster and pre-select the 10 words with the highest score in each cluster. We call this as local candidate selection, since it is done independently for each cluster. For each of the 10 selected terms we compute information gain IGm, [6]. The three terms in each cluster with the highest score are chosen as candidates. We call this as global candidate selection, because the computation of IGm for a term in a cluster is dependent also on the contents of the other clusters. Global selection has the purpose of obtaining different labels for different clusters. At the end of this procedure, if two clusters have the same signature we merge them.

**(6) Second-level clustering:** For second-level clustering we adopt a different approach, since metric-based clustering applied at the second level tends to detect a single large “inner core” cluster and several small “outlier” clusters. The second-level part of the hierarchy is generated based on the candidate words found for each cluster during the first-level candidate words selection. Calling $K$the set of three candidate words of a generic cluster, we consider all its subsets as possible signatures for second level clusters.

## 4 Experimental Evaluation

In this paper, precision is used to take into account both relevance and membership degrees. Since all three algorithms tested using overlapping clusters, modifying the weight of a result according to its membership degree is used to prevent variation of precision due to the same result being present in multiple clusters. M-FPF employs clusters which record membership degrees in the range [0, 1], while STC and Lingo appear to employ overlapping clusters, which does not define membership degree. In this case, membership degree is set to the inverse of the number of clusters of which a result is a member. M-FPF records relevance in the range [0, 1], however STC and

Lingo do not use relevance or sort results in any fashion, so they only use precision weighted by membership degrees in the tests that follow.

The other key problem is the data to be used for testing. For testing, sets of search results are downloaded and saved so they are identical between runs. Each dataset consists of 100 results, each with a title, snippet and URL. As the following results show, the algorithms' performance depends heavily on the dataset and its distribution of search results. This paper includes the results of M-FPF, STC and Lingo on four queries used in other papers [3], [8] (Jaguar, Apple, Java, Salsa), using the Google! search API.

### 4.1 Clustering Quality

Search Engine Results In all the tests that follow, the parameters of the three algorithms are left unchanged between runs. A fixed number of six clusters was chosen for all datasets, as there are at least ten topic labels and generation of too many clusters for a relatively small number of results can result in excessive fragmentation of categories. STC and Lingo were left to the defaults set by the Carrot2 software, M-FPF has the following default parameters set, unless otherwise noted: Nc = 10, the parameter is chosen to give on average a balance between precision and recall. The graphs show three bars for each of the algorithms: on the left is weighted precision, the middle is recall and on the right is a relevance score. A difference between precision and recall indicates a tendency of an algorithm to return only a small number of results that have a high probability of being correctly classified (high precision, low recall) or a large number of results in each cluster, with high overlap between clusters (low precision, high recall).

Figure 2 shows the performance of the three algorithms on two datasets: Jaguar and Apple. These two datasets are a fairly average case with three or four large clusters and two or three smaller clusters with low to moderate overlap between the clusters. M-FPF performs well in these cases, delivering a balance between precision and recall. All three algorithms show higher precision in the Jaguar dataset and higher recall in the Apple dataset, possibly indicating higher overlap in the later.

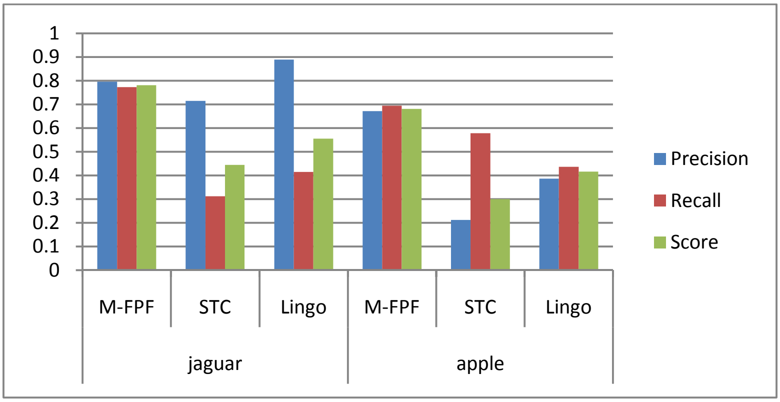

**Fig. 2.** Cluster quality measured using precision, recall and R1 score for the Jaguar and Apple datasets

M-FPF and Lingo was able to extract almost all of the major topics, while STC was unable to extract more than half. Lingo suffered from a low number of classified results (many results were binned in 'Other Topics' i.e. as outliers), which is expected from its design focus on cluster purity [4].

Figure 3 shows the performance of the three algorithms on two more challenging datasets: Java and Salsa. Both datasets are dominated by two large clusters and four or five smaller clusters, making it hard for the algorithms to effectively extract the topics. As a result, recall and overall score of all three algorithms drop significantly. The Salsa dataset also has very high overlap, and due to the default parameters for M-FPF, it performs similarly to Lingo while STC performs slightly better overall.

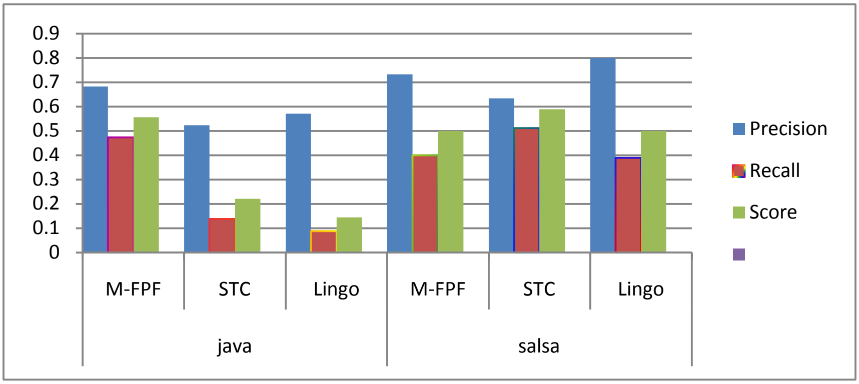

**Fig. 3.** Cluster quality measured using precision, recall and R1 score for the Java and Salsa datasets

## 5 Conclusion

This paper demonstrated M-FPF, a web search result clustering and the labeling tasks are performed on the fly by processing only the snippets provided by the auxiliary search engines, and use no external sources of knowledge. Clustering is performed by means of a modified version of the furthest-point-first algorithm.. Finally, M-FPF performs well compared to the established STC and Lingo algorithms, demonstrating both good quality clustering without using a more complex label driven approach to document clustering. Enhancing the performance of search engines and improving the usability of search results is an active area of research, and clustering web search results is only one way of doing this. however, improvements can be made its efficiency as well as the use of hierarchies to improve document organization.

## References

1. Zamir, O., Etzioni, O.: Web document clustering: A feasibility demonstration. In: Proceedings of the 21st Annual International SIGIR Conference on Research and Development in Information Retrieval (1998)

2. Hanumanthappa, M., Prakash, B.R., Mamatha, M.: Improving the efficiency of document clustering and labeling using Modified FPF algorithm. In: Proceeding of International Conference on Problem Solving and Soft Computing (2011)
3. Geraci, F., Leoncini, M., Montangero, M., Pellegrini, M., Renda, M.E.: *FPF-SB*: A Scalable Algorithm for Microarray Gene Expression Data Clustering. In: Duffy, V.G. (ed.) HCII 2007 and DHM 2007. LNCS, vol. 4561, pp. 606–615. Springer, Heidelberg (2007)
4. Osinski, S., Weiss, D.: A concept-driven algorithm for clustering search results. IEEE Intelligent Systems 20(3), 48–54 (2005)
5. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the 34th Annual ACM Symposium on the Theory of Computing, STOC 2002, Montreal, CA, pp. 380–388 (2002)
6. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the14th International Conference on Machine Learning, ICML 1997, Nashville, US, pp. 412–420 (1997)
7. Ferragina, P., Gulli, A.: A personalized search engine based on Web-snippet hierarchical clustering. Special Interest Tracks and Poster Proceedings of the 14th International Conference on the World Wide Web, WWW 2005, Chiba, JP, pp. 801–810 (2005)
8. Crabtree, D., Gao, X., Andreae, P.: Standardized evaluation method for web clustering results. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (2005)
9. Matsumoto, T., Hung, E.: Fuzzy Clustering and Relevance Ranking of Web Search Results with Differentiating Cluster Label Generation
10. Geraci, F., Pellegrini, M., Maggini, M., Sebastiani, F.: Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution. In: Crestani, F., Ferragina, P., Sanderson, M. (eds.) SPIRE 2006. LNCS, vol. 4209, pp. 25–36. Springer, Heidelberg (2006)
11. Geraci, F., Pellegrini, M., Pisati, P., Sebastiani, F.: A scalable algorithm for high-quality clustering of Web snippets. In: Proceedings of the 21st ACM Symposium on Applied Computing, SAC 2006, Dijon, FR, pp. 1058–1062 (2007)
12. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science 38(2/3), 293–306 (1985)