International Journal of Engineering Innovation & Research
Volume 3, Issue 3, ISSN: 2277 – 5668

Personalized Concept Oriented Search Engine with User Profiling and Result Preference Based on Lucene

Devu. M
Assistant Professor
Department of Computer Science and Engineering
Mohandas College of Engineering and Technology, India
Email: alwaysdevu@yahoo.co.in

Abstract **– The fundamental component of any personalization application is user profiling. A good user profiling strategy is an essential and fundamental component in search engine personalization. Evaluating user preferences of web search results is crucial for search engine development, deployment and maintenance. This paper focuses on search engine personalization and concept-based user profiling that are based on both positive and negative preferences of the users. The underlying idea of our proposed system is based on concepts and their relations extracted from the submitted queries, the Web-snippets and the click through data. Incorporating user behaviors data can significantly improve ordering of top results in web searching. The relationships between users can be mined from the concept-based user profiles to perform collaborative filtering.This allows users with the same interests to share their profiles.Finally,the concept-oriented user profiles can be integrated into the ranking algorithms of search engine so that search results can be ranked according to individual user's interest.**

Keywords **– User Profiling, Personalization, Concept, Lucene, Web-Snippets, Web Search, Click Through, Search Engine.**

I. INTRODUCTION

Data mining, a branch of computer science and artificial intelligence, is the process of extracting patterns from data. Data mining is seen as an increasingly important tool by modern business to transform data into business intelligence giving an informational advantage. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery. The related terms data dredging, data fishing and data snooping refer to the use of data mining techniques to sample portions of the larger population data set that are (or may be) too small for reliable statistical inferences to be made about the validity of any patterns discovered. These techniques can, however, be used in the creation of new hypotheses to test against the larger data populations. As data mining can only uncover patterns already present in the data, the target dataset must be large enough to contain these patterns while remaining concise enough to be mined in an acceptable timeframe. A common source for data is a data mart or warehouse. The various processes include preprocessing, mining & validation

Pre-process is essential to analyze the multivariate datasets before clustering or data mining. Data mining commonly involves four classes of tasks:

- Clustering- is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
- Classification- is the task of generalizing known structure to apply to new data. Common algorithms include decision tree learning, nearest neighbor, naive Bayesian classification, neural networks and support vector machines.
- Regression - Attempts to find a function which models the data with the least error.
- Association rule learning -Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

The final step of knowledge discovery from data is to verify the patterns produced by the data mining algorithms occur in the wider data set. Not all patterns found by the data mining algorithms are necessarily valid. It is common for the data mining algorithms to find patterns in the training set which are not present in the general data set, this is called over fitting. To overcome this, the evaluation uses a test set of data which the data mining algorithm was not trained on. The learnt patterns are applied to this test set and the resulting output is compared to the desired output.

A web search engine is designed to search for information on the World Wide Web and FTP servers. The search results are generally presented in a list of results and are often called hits. The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. As the Web keeps expanding, the number of pages indexed in a search engine increases correspondingly. With such a large volume of data, finding relevant information satisfying user needs based on simple search queries becomes an increasingly difficult task.

The specific contribution of this paper includes

- The analysis of alternatives for incorporating user behaviors into web search ranking. First, relationships between users can be mined from the concept-based user profiles to perform collaborative filtering. This allows users with the same interests to share their profiles.
- Second, the existing user profiles can be used to predict the intent of unseen queries, such that when a user

submits a new query, personalization can benefit the unseen query.

- Finally, the concept-based user profiles can be integrated into the ranking algorithms of a search engine so that search results can be ranked according to individual users' interests. The main features includes

Concept based profiling

It aims at capturing user's conceptual needs. User's browsed documents and search histories are automatically mapped into a set of topical categories [8].

Heterogeneous profiling

Single large user profile for each user in the personalization process.

Our approach consists of the following four major steps. First, when a user submits a query, concepts (i.e., important terms or phrases in web-snippets) and their relations are mined online from web-snippets to build a concept relationship graph. Second, click through are collected to predict user's conceptual preferences. Third, the concept relationship graph together with the user's conceptual preferences is used as input to a concept-based clustering algorithm that finds conceptually close queries. Finally, the most similar queries are suggested to the user for search refinement.

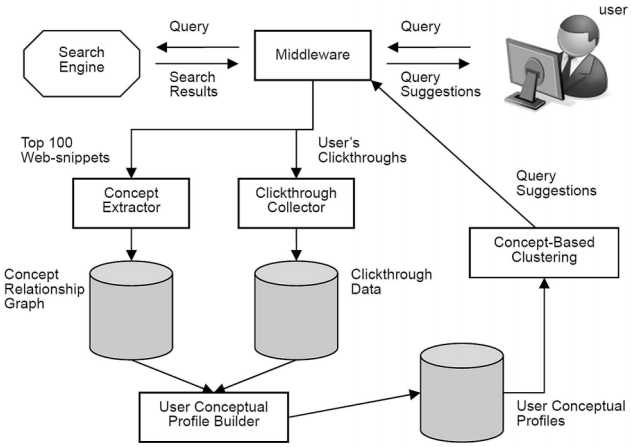

Fig.1. System Architecture and System Design Model

II. MODULES AND DESCRIPTIONS

2.1 Search module

A web search engine is designed to search for information on the World Wide Web and FTP servers. The search results are generally presented in a list of results and are often called hits. The information may consist of web pages, images, information and other types of files.

A search engine operates, in the following order

- Web crawling
- Indexing
- Searching

Web search engines work by storing information about many web pages, which they retrieve from the html itself. These pages are retrieved by a Web crawler (sometimes also known as a spider) — an automated Web browser which follows every link on the site. The contents of each page are then analyzed to determine how it should be indexed (for example, words are extracted from the titles, headings, or special fields called meta tags). Data about web pages are stored in an index database for use in later queries. A query can be a single word.

The purpose of an index is to allow information to be found as quickly as possible. The user normally expects the search terms to be on the returned pages. Increased search relevance makes these cached pages very useful, even beyond the fact that they may contain data that may no longer be available elsewhere.

2.1.1 Normal Search

When a user enters a query into a search engine (typically by using key words), the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. The index is built from the information stored with the data and the method by which the information is indexed. The engine looks for the words or phrases exactly as entered and lists all the pages which contain that words or phrases.

2.1.2 Profile Based Search

A user should register before making the searching process. Once the user is registered the profile of the user is created. For the purpose of searching user have to login. The index is built from the information stored with the data and the concept preference of the user. It also takes into account about the user's positive preference and negative preference. When a user enters a query, a log will be maintained with the query and result pair along with date. Next time, if similar information of previous search is given its results are loaded swiftly with the log of the previous search.

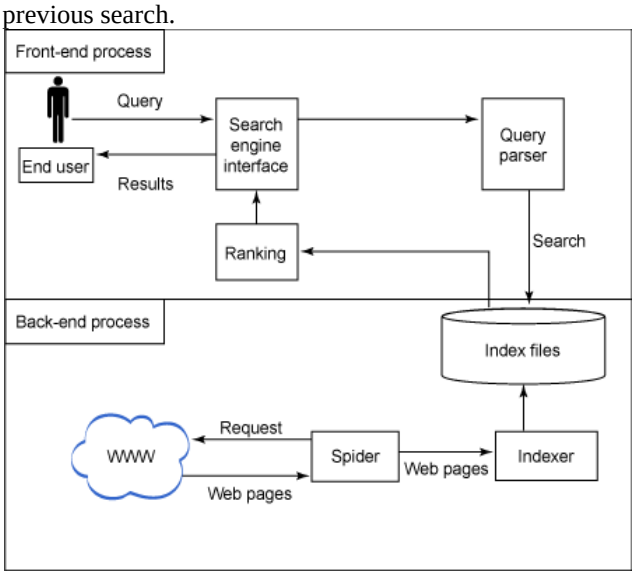

Fig.2. Web search engine architecture

The architecture of a common Web search engine contains a front-end process and a back-end process [12]. In the front-end process, the user enters the search words into the search engine interface, which is usually a Web page with an input box. The application then parses the search request into a form that the search engine can understand, and then the search engine executes the search

operation on the index files. After ranking, the search engine interface returns the search results to the user. In the back-end process, a spider or robot fetches the Web pages from the Internet, and then the indexing subsystem parses the Web pages and stores them into the index files.

2.2 Concept Mapping Module

When the user searches for the query the concept space derived from web snippets containing the concepts are displayed. For example. When the user searches for the query "apple" the concept space derived from web snippets contains concepts such as "ipod", "iphone" and recipes. If the user is indeed interested in the concept "recipes" and clicks on pages containing the concept "recipe", the click through should gradually favour the concept "recipe" by assigning higher weights to the nodes, but the weights of unrelated topics such as iphone ,ipod should have a weight zero.

Concept Extraction

Our concept extraction method, which is composed of the following three basic steps:

1) Extracting concepts using the web-snippets returned from the search engine,

2) Mining concept relations, and

3) Creating a user concept preference profile using the extracted concepts, concept relations and user's click thorough.

Extracting Concepts from Web Snippets

After a query is submitted to a search engine, a list of web snippets is returned to the user. If a keyword or phrase exists frequently in the web snippet of a particular query it represents an important concept related to the query. To extract concepts for a query q, we first extract all the keywords and phrases from the web-snippets returned by the query[1].

Mining Concept Relation

We assume that two concepts from a query q are similar if they coexist frequently in the Web-snippets arising from the query q. Fig. 5. An example of a concept space and the corresponding user profile. (a) The concept space derived for the query "apple." (b) An example of user profile in which the user is interested in the concept "Macintosh."

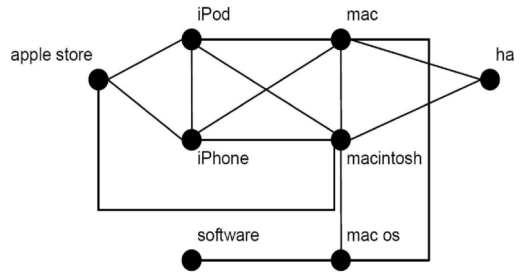

Fig.3a. Concept space for the corresponding user profile

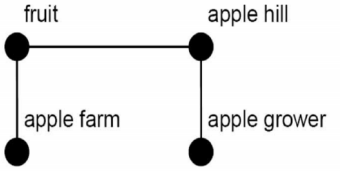

Fig.3b. shows a concept graph built for the query "apple."

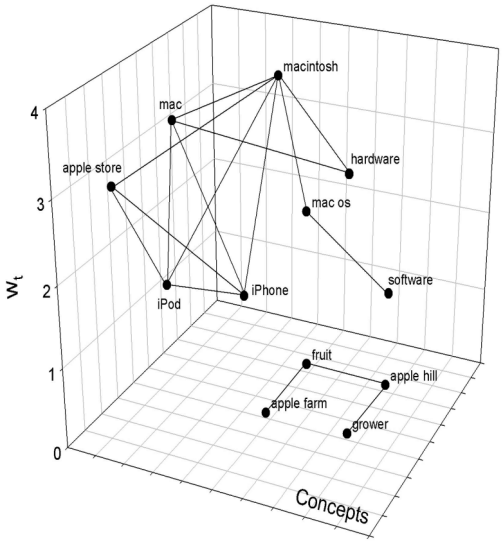

Fig.3c. An example of user profile in which the user is interested in the concept "Macintosh."

Personalized Concept based clustering

Personalized effect is achieved by manipulating the user concept preference profile in the clustering process. Our personalized concept-based clustering algorithm [6] with which ambiguous queries can be classified into different query clusters [9]. Concept-based user profiles are employed in the clustering process to achieve personalization effect. If two given queries whether they are identical or not means different things to two different users, they should not be merged together because they refer to two different sets of concepts for the two users[4].

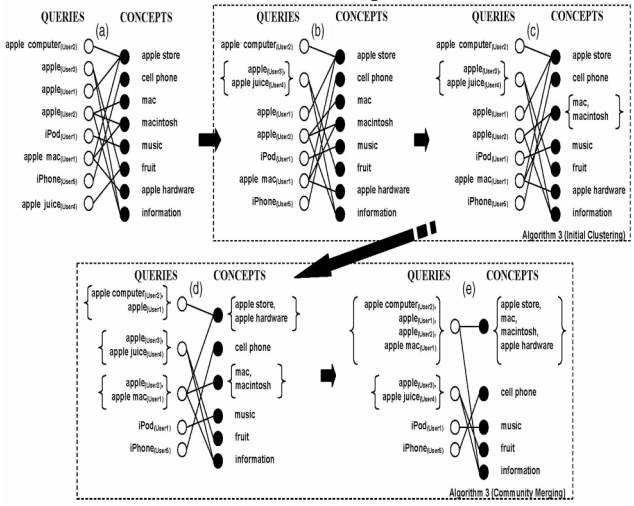

Fig.4. Performing personalized concept-based clustering algorithm on a small set of click through data. Starting from top left:
(a) The original bipartite graph. (b), (c) Initial clustering. (d), (e) Community merging.

2.3 Ranking with result preference

Ranking search results is a fundamental problem in information retrieval. Modern web search engines rank results based on a large number of features, including content-based features (i.e., how closely a query matches

the text or title or anchor text of the document), and query-independent page quality features (e.g., Page Rank of the document or the domain) [1]. In most cases, automatic (or semiautomatic) methods are developed for tuning the specific ranking function that combines these feature values [2].

Our main approach is to model user web search behavior as if it were generated by two components: a "relevance" component – query-specific behavior influenced by the apparent result relevance according to a specific user and a "background" component [3] – users clicking indiscriminately. Our general idea is to model the deviations from the expected user behavior. The features we use to represent user search interactions can be grouped into Query – text, Click through, and browsing [5].

Each clicks of the user are tracked and the time that each user spend on a particular web page is monitored. Depending on the click and the time spent on each, the concept of the user is mined and extracted. when the actual interest of the user is obtained, the web search result for a specific query is delivered to user based on ranking [7].

2.4 Analysis and Configuration of LUCENE

Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform. Lucene, a full-text information retrieval (IR) library written in the Java language [11]. You can embed Lucene easily into your applications and implement indexing and searching functionality. Now it's an open source project in the popular Apache Jakarta Project family[13]. Lucene itself is just an indexing and search library and does not contain crawling and HTML parsing functionality. However, several projects extend Lucene's capability: Apache Nutch provides web crawling and HTML parsing. Lucene has many features. It

- Has a powerful, accurate, and efficient search algorithm.
- Calculates a score for each document that matches a given query and returns the most relevant documents ranked by the scores.
- Supports many powerful query types, such as Phrase Query, Wildcard Query, Range Query, Fuzzy Query, Boolean Query, and more.
- Supports parsing of human-entered rich query expressions.
- Allows users to extend the searching behavior using custom sorting, filtering, and query expression parsing.

2.4.1 Steps in building applications using Lucene

Building a full-featured search application using Lucene primarily involves indexing data, searching data, and displaying search results.

2.4.2 Indexing the Lucene architecture

Lucene uses different parsers for different types of documents. Take HTML documents, for example -- an HTML parser does some preprocessing, such as filtering the HTML tags and so on[13]. The HTML parser outputs the text content, and then the Lucene Analyzer extracts tokens and related information, such as token frequency, from the text content. The Lucene Analyzer then writes the tokens and related information into the index files of Lucene.

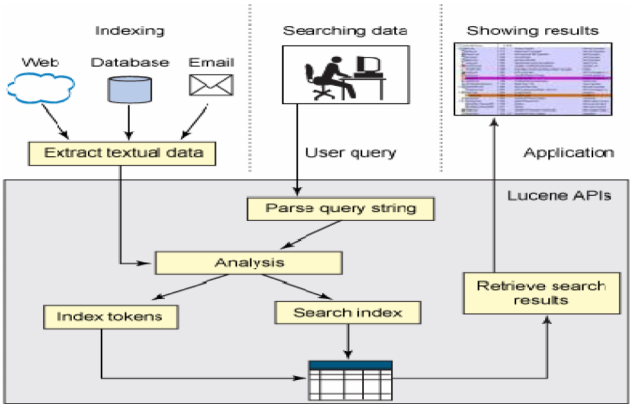

Fig.4. Steps in building applications using Lucene

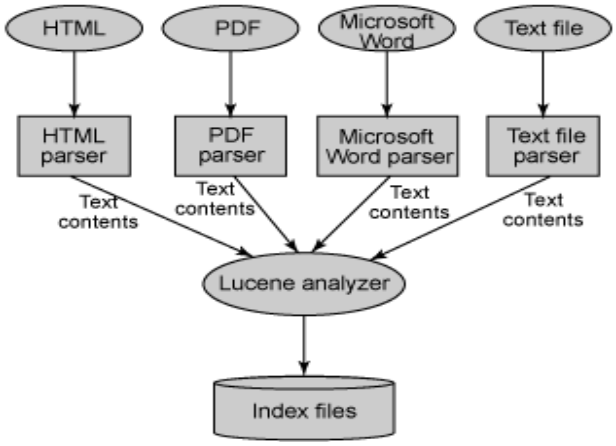

Fig.5. indexing the Lucene architecture

III. CONCLUSION

An accurate user profile can greatly improve a search engine's performance by identifying the information needs for individual users [10]. The search is based on concept preference of the user .The user's positive preference and negative preferences are taken into account. The profile based user when submits a query will obtain a web page which is of users interest without any delay. This is done using a ranking system, which considers user's concept preference. Separate profiles are present for each user. The user is also able to view the previously searched information's with the help of logs, which is evaluated during the search process. The users with the same interest are allowed to share their profiles.

FUTURE ENHANCEMENT

As a future work, the existing user profiles can be used to predict the intent of unseen queries, such that user when a user submits a new query, personalization can benefit the unseen query. For the first time, when the user submits a new query the system should be able to produce the result what the user wish to obtain.

REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais, "Improving Web Search Ranking by Incorporating User Behavior Information," *Proc. ACM SIGIR*, 2006.
[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, "Learning User Interaction Models for Predicting Web Search Result Preferences," *Proc. ACM SIGIR*, 2006.
[3] R.Baeza-yates,C.Hurtado,and M. Mendoza, "Query Recommendation Using Query Logs in Search Engines," *Proc. Int'lWorkshop Current Trends in Database Technology, pp. 588-596*, 2004.
[4] D.Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log," *Proc. ACM SIGKDD*, 2000.
[5] T. Joachims, "Optimizing Search Engines Using Clickthrough data," *Proc. ACM SIGKDD*, 2002.
[6] K.W.-T. Leung, W. Ng, and D.L. Lee, "Personalized Concept-Based Clustering of Search Engine Queries," *IEEE Trans. Knowledge and Data Eng., vol. 20, no. 11, pp. 1505-1518*, Nov. 2008.
[7] F. Liu, C. Yu, and W. Meng, "Personalized Web Search by Mapping User Queries to Categories," *Proc. Int'l Conf. Information and Knowledge Management (CIKM*), 2002.
[8] M. Speretta and S. Gauch, "Personalized Search Based on User Search Histories," *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence*,2005.
[9] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Query Clustering Using User Logs," *ACM Trans. Information Systems, vol. 20, no. 1, pp. 59-81*, 2002.
[10] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search," *Proc. World Wide Web (WWW) Conf.*,2007.
[11] http://www.ibm.com/developerworks/web/library/wa-lucene2
[12] http://computer.howstuffworks.com/internet/basics/search-engine1.htm
[13] http://www.javaranch.com/journal/2004/04/Lucene.html

AUTHOR'S PROFILE

Devu.M

was born in Kerala, India on May 18 1987. I am a first class M.Tech holder in Computer Science and Engineering. My special fields of interest include Big data analytics and Datamining in cloud. In future work, I plan to build on my experience in large-scale data analytics, and to expand my work to other areas of computer systems and cloud computing .One approach I am currently exploring is to leverage the deterministic nature of most cluster computing models to let users selectively replay part of the computation at low cost. Overall, my philosophy is to find the fundamental problems in new systems, and search for highly practical solutions. The combination of rapidly emerging problems and friendliness to open source allows for rich, intellectually rewarding research that can directly shape the platforms powering some of today's most exciting applications.