Interactive Guided Online/Off-line search using Google API and JSON

Kalyan Netti[1]
[1] Scientist, NGRI(CSIR),
Hyderabad, Andhra Pradesh, India

Abstract

Information retrieval systems (e.g., web search engines) are critical for overcoming information overload. A major deficiency of existing retrieval systems is that they generally lack user modeling and are not adaptive to individual users, resulting in inherently non-optimal retrieval performance [1]. Sources of these problems include the lack of support for query refinement. Web search engines typically provide search results without considering user interests or context. This in turn increases the overhead on the search engine server. To address these issues we propose a novel interactive guided Online/off-line search mechanism. The system allows user to choose for normal or combinational search [5] of the query string and allows the user to store the best search results for the query string. The proposed system also provides option for off-line search which searches from the bundle of stored results. Systems which implemented offline search require downloading and installing the stored bundle of search results before using it. The proposed system is an interactive web based search facility both offline and online. The system doesn't require installing the bundle of saved search results for offline searching, as the search results are added to the bundle interactively as chosen by the user. The system is very likely to return the best possible result as it uses combinational search. The result from the combination search can be stored and can be searched again offline. Experiments revealed that combination search of keywords in query yields variety of results. Thus the Bundle of Stored result consists of best possible results as the user chooses to save in it. This will enhance the systems searching capabilities offline, which in turn reduces the burden on the search engine server.

Keywords: *Web search Engine, Meta-search engine***,** *Information retrieval, Google, Retrieval Models, Offline Search, Combination Search, Google API, JSON*

1. Introduction

One of the most pressing issues with today's explosive growth of the Internet is the so-called resource discovery problem[3]. Although many information retrieval systems have been successfully deployed, the current retrieval systems are far from optimal [1]. A major deficiency of existing retrieval systems is, they generally lack user modeling and are not adaptive to individual users [4]. This non-optimality or inconvenience to the users may sometimes be seen as the search engine inability to understand the user query, but if it is seen through user perspective, two specific issues can be observed.

a) Different users may use a search query in many different ways, but clearly thinking of one result. Like, a user may give a query as “java programming”, other user may give the query like “programming java”. Sometimes the search engine may give different results even though the user perspective is same. b) User after giving the query has to go through the information of all websites which are returned by the search engine, even though a best search engine can filter the non-query related sites. This process although a common one is very tedious. Once the user is not satisfied with the result he/she may repeat the process with a different query possibly with a different combination of keywords. This may overburden the search server with repeated queries.

A system which can give user an option for searching with different combinations of keywords in the query is a viable solution for the problem mentioned at ‘a’ above. Also a system which allows the user to save the best possible results as a bundle, and accessing them later offline is a probable solution for ‘b’ mentioned above. The bundle of search results saved can be searched again for further refinement by any other user for a same type of query or combination.

The system implemented in this paper is an interactive interface and provides an option for the user to search the server with a combination of keywords or with a normal search. It allows the user to save the search result as a bundle of website URLs. Later users can access the bundle offline for further refinement of the search. This will reduce overhead on the network and on the server when the search is performed for the same type of result. Also the result can be assured of the best result as it is yielded from the combination of keywords. The main aim of this paper is to make information search handy and appropriate such that it will be easy to use while offline thus reducing the overhead on the search engine server. The system uses Google as a search server and to refine the URLs of the search results JSON and Google API is used. The reason for using Google as a search server is that, Google has grown to be one of the most popular search engines that are available on the Web. Like most other search engines, Google indexes Web sites, Usenet news groups, news sources, etc. with the goal of producing search results that are truly relevant to the user. This is done using proprietary algorithms, which work based on the understanding that if a page is useful, other pages covering the same topic will somehow link to it. So, it can be said that Google focuses on a page's relevance and not on the number of responses [6]. There for it can be said that Google focuses on a page's relevance and not on the number of responses.

Moreover, Google allows sophisticated searches, with required and forbidden words, and the ability to restrict results based on particular language or encoding [8]. However, only a small number of web users actually know how to utilize the true power of Google. Most average web users, make searches based on imprecise query keywords or sentences, which presents unnecessary, or worse, inaccurate results to the user. Based on this assumption, applications that help guide user's searching sessions have started to emerge. This is further motivated by the introduction of Google Web Services, which allows developers to query the Google server directly from their application [5] [8].

Google has been providing access to its services via various interfaces such as the Google Toolbar and wireless searches. And now the company has made its index available to other developers through a Web services interface. This allows the developers to programmatically send a request to the Google server and get back a response. The main idea is to give the developers access to Google's functionality so that they can build applications that will help users make the best of Google. The Web service offers existing Google features such as searching, cached pages, and spelling correction [7]. It is provided via Google AJAX Search API and can be manipulated in any way that the programmer pleases. Also Google is now allowing queries using a REST-based interface that returns search results using JSON .

2. Related Work

Great deal of work has been done in making available guided search such as GuideBeam [10], which is the result of research work carried out by the DSTC (Distributed Systems Technology Centre) at the University of Queensland in Brisbane, Australia. GuideBeam works based on a principle called "rational monotonicity" that emerged from artificial intelligence research in the early nineties. In the context of GuideBeam, rational monotonicity prescribes how the user's current query can be expanded in a way which is consistent with the user's preferences for information. In other words it is a guiding principle of preferential reasoning [11]. Since users prefer certain pieces of information in their quest for information, preferential reasoning fits very nicely into the picture of guided searching. Users can intuitively navigate to the desired query in a context-sensitive manner. This is known as "Query by Navigation". The goal is to elicit a more precise query from the user, which will translate into more relevant documents being returned from the associated search engine. Another example that is more closely related to Google would be the Google API Search Tool by Softnik Technologies [12]. It is a simple but powerful Windows software tool for searching Google. It is completely free and is not meant to be a commercial product. All that the users need to do is register with Google for a license key and they will be entitled to pose 1000 queries a day. It is also an efficient research tool because it allows the users to record the search results and create reports of their research easily and automatically. The URLs, titles, etc. can be copied to the clipboard and then to a spread sheet or any other software. In summary, it enables the users to organize and keep track of their searching sessions, all at the convenience of their desktops.

The Google API Search Tool requires the users to download and install the software before they can start using it. An alternative is to have a Web-based version of the search tool. Many projects have been exploring this path like [5] Guided Google, Google API Proximity Search (GAPS) developed by Staggernation.com [13]. The GAPS is developed using Perl and uses the Google API to search Google for two search terms that appear within a certain distance from each other on a page [6]. Most of the Frameworks mentioned above are either free tools available or commercial ones. Most of the tools mentioned above like Guided Google, Staggernation etc were developed using Google SOAP API, which is discontinued by Google and those tools are no longer available on web. Most of the tools are not interactive. Some systems like COS [14] have the limitation of downloading and installing the COS pack [14] offline and it has no facility of Combination Search, thus again resulting in inconvenience to the user.

The main difference between the tools and systems mentioned above and the system proposed in this paper is that there is no need for the user to download the bundle of best search results for offline search just as implemented in COS [14].Instead, user has an option to choose between offline and online search. The system allows allow users to save beset results interactively. The system has the facility for combination search which proved one to retrieve best results. The system is implemented as web interface and is easy to use. It uses powerful Google Ajax API which is the latest search API from Google [7], JSON for parsing the streamed response. Thus the proposed system provides a focused search and is very likely to return the relevant response to user.Online help for parsing the streamed response using JSON through web interface by using JSP and Servlets is not available in internet although help for stand-alone is available.

3. Architecture

3.1 Proposed System Architecture

Figure 1 shows the overview and architecture of the proposed system. The user access the system as a web interface which is developed using JSPs. The JSPs are

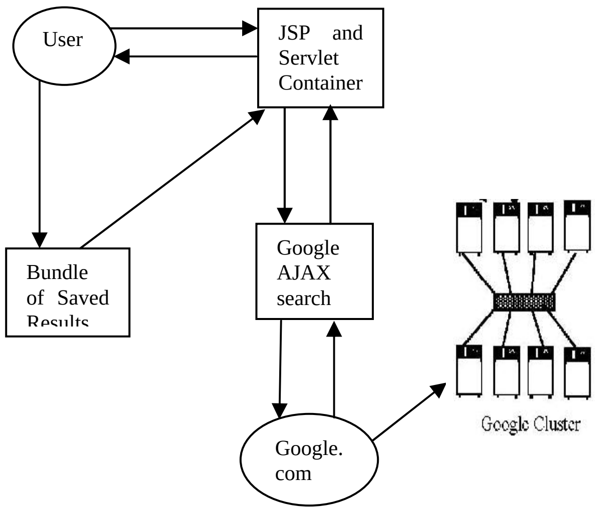


Figure 1: Main Architecture

hosted on a Tomcat Server. The Tomcat server acts as both a stand-alone webserver and servlet container. Before submitting the Query, the user has the option to choose between online and offline search. If the user chooses online search then the system will ask to choose either Combination Search or Normal Search. The search is then handled by the Google AJAX Api and is sent to Google.com. Google has Google cluster which is further divided into smaller clusters that is made up of a few thousand machines, and is geographically distributed to protect Google against catastrophic data centric failures. Each of these smaller clusters will be assigned queries based on the user's geographic proximity to it [15, 16]. This is a form of load balancing, and it helps to minimize the round-trip time for a query; hence, giving greater response time. If the user chooses for online search, the response is handled by the JSON implemented in JSP/Servlet Container in Tomcat and the search result is displayed to the User. The user has then an option to save the search results directly to the bundle of saved results. If the user chose for the offline search then the JSP/Servlet container sends the request to the Bundle of Saved Results and returns the response to the user.

3.2 Google Ajax Search API

Figure 2, shows the Google API architecture. The architecture of how Google Web Services interact with user applications is shown in Figure 2. The Google server is responsible for processing users' search queries [9]. The programmers develop applications in a language of their choice (Java, C, Perl, PHP, .NET, etc.) and connect to the remote Google Web APIs service. Communication is performed via the Google AJAX Search Api. The AJAX Search API allows you to easily integrate some very powerful and diverse Google based search mechanisms or "controls" onto a Web page with relatively minimal coding. These include: [7]

a) Web Search: This is a traditional search input field where, when a query is entered, a series of text search results appear on the page.

b) Local Search: With Local Search, a Google Map is mashed together with a search input field and the search results are based on a specific location.

c) Video Search: The AJAX Video Search provides the ability to offer compelling video search along with accompanying video based search results.

Once connected, the application will be able to issue search requests to Google's index of more than two billion web pages and receive results as structured data, access information in the Google cache, and check the spelling of words. Google Web APIs support the same search syntax as the Google.com site.

In short, the Google AJAX APIs serve as an enhanced conduit to several of Google's most popular hosted services. The hosted services such as Google Search or Google Maps can be accessed directly, but with AJAX APIs comes the ability to integrate these hosted services into anyone's custom web pages. The way the AJAX APIs work is by allowing any web page that is hosted on the Internet access to Google search (or feed) data through JavaScript code.

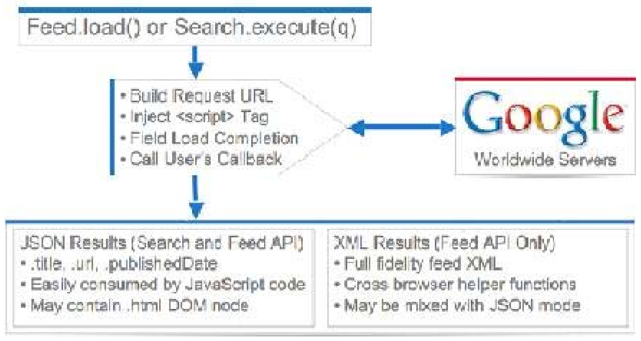
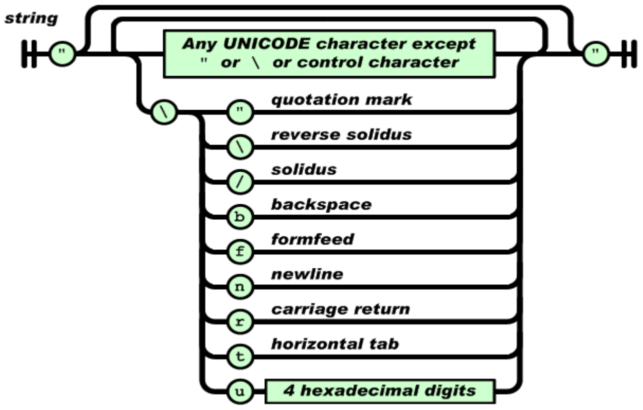

Feed.load() or Search.execute(q)
• Build Request URL
• Inject <script> Tag
• Field Load Completion
• Call User's Callback
Google
Worldwide Servers
JSON Results (Search and Feed API)
• .title, .url, .publishedDate
• Easily consumed by JavaScript code
• May contain .html DOM node
XML Results (Feed API Only)
• Full fidelity feed XML
• Cross browser helper functions
• May be mixed with JSON mode

Figure 2: Google AJAX Search API Architecture

The core JavaScript code that fetches the search or feed data can be as simple as Search.execute( ) or Feed.load( ). As the request is made to Google's worldwide servers, a response of either Search data or prepared AJAX Feed data is streamed back to the Web page in either JSON (JavaScript Object Notation) or XML formats. Parsing of this data can either be done manually or automatically by using one of the provided UI controls that are built upon the lower level AJAX APIs.

3.3 JSON

JSON (an acronym for JavaScript Object Notation) is a lightweight text-based open standard designed for human-readable data interchange. It is derived from the JavaScript programming language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for virtually every programming language.
In JSON the String data structure take on these forms



JSON Schema [17] is a specification for a JSON-based format for defining the structure of JSON data. JSON Schema provides a contract for what JSON data is required for a given application and how it can be modified, much like what XML Schema provides for XML. JSON Schema is intended to provide validation, documentation, and interaction control of JSON data. JSON Schema is based on the concepts from XML Schema, RelaxNG, and Kwalify, but is intended to be JSON-based, so that JSON data in the form of a schema can be used to validate JSON data, the same serialization/deserialization tools can be used for the schema and data, and it can be self descriptive.
Apart of certain limitations which are limited to textual data formats which also apply to XML and YAML, JSON is primarily used for communicating data over the Internet. The proposed system in this paper uses JSON extensively to parse the response send by the Google API. Parsing the response using JSON enables the system to store the results in array, get the required URLs, Count the total results etc. This enables the user to compare the query string during combinational search, by total number of results retrieved and save the best results.

4. Design and Implementation

The Proposed system is an interactive web interface, which is developed using JAVA and coded in JSP and Servlets. As discussed earlier, this system is developed based on the assumption that search engines typically provide search results without considering user interests or context, which in turn leads to Overhead of the search engine server.
This implementation tries to demonstrate, how the new Google AJAX search API can be fully utilized, how it helps the users to guide for better results, how it reduces the overhead on the search engine sever.
This system can be grouped into four categories. The first one is for choosing between Online and offline search. The second one is choosing normal search or combination search. The third one is saving the results into the bundle of storage. The fourth is searching the bundle of stored results offline. Each of these is discussed in detail in following sections. A simple illustration of how this system works is given at figure 3;

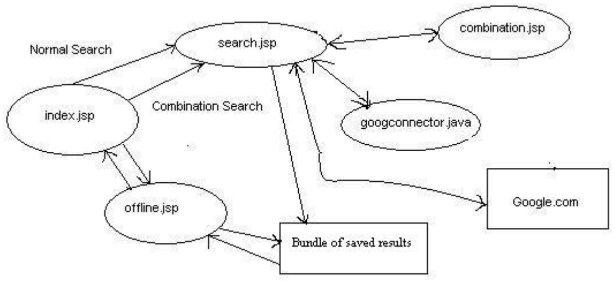


Figure 3: Design and Implementation of the system

4.1 Online Search

There is a main page (index.jsp) that is used to interface with all other files. This page consists of query text box (a shown in figure 4), and options for the user to Offline search and online search. If the user chooses online search the page display the option to choose for normal search or combination search. The normal and combination search functions are supported by search.jsp program file. The program file search.jsp connects to Google.com by using googconnector.java bean which establishes the connection using Google AJAX Api.

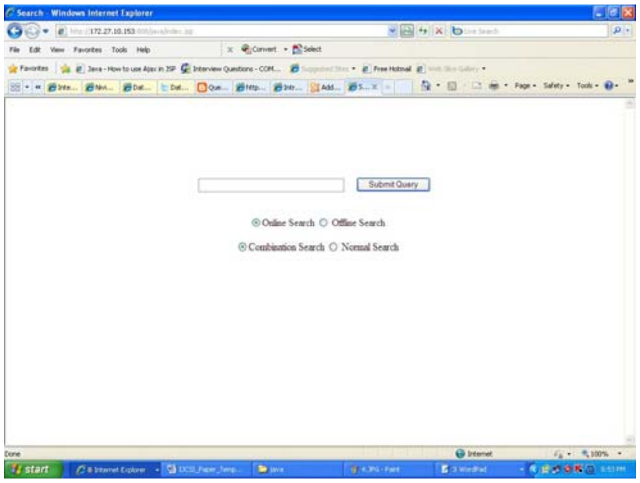


Figure 4: Online Search and Combination Search

4.2 Combination Search

As mentioned earlier this system provides a function that will automatically calculate the permutation and make different combinations of the keywords used. For example, if the users were to key in the term “java programming”, it will automatically generate two combinations, which are “java programming” and “programming java”. The results of these two queries when searched using Google.com are very different (see Figure 5 and Figure 6).Generating different combinations of key words in query string is carried out by combination.jsp program file.

4.3 Parsing Response

Once the response is send by the Google, search.jsp parses the streamed response using JSON. The code snippet for connecting to Google.com using AJAX API Search is as follows

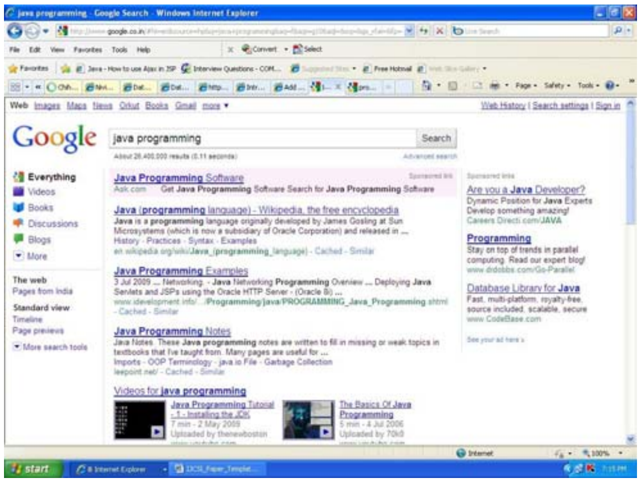

java programming - Google Search - Windows Internet Explorer
Google
java programming
Search
Java Programming Software
Java (programming language) - Wikipedia, the free encyclopedia
Java Programming Examples
Java Programming Notes
Videos for java programming
Are you a Java Developer?
Programming
Database Library for Java

Figure 5: searching google.com using “java programming”

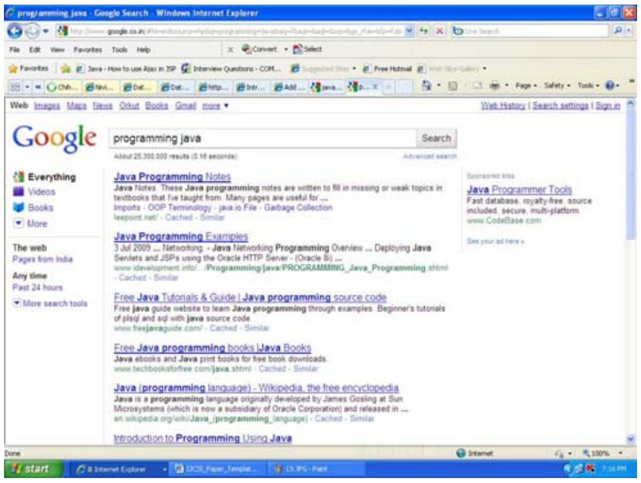

programming java - Google Search - Windows Internet Explorer
Google
programming java
Search
Java Programming Notes
Java Programming Examples
Free Java Tutorials & Guide | Java programming source code
Free Java programming books |Java Books
Java (programming language) - Wikipedia, the free encyclopedia
Introduction to Programming Using Java
Java Programmer Tools

Figure 6: searching google.com using “programming java”

```
// Encode the Query to make a valid one
//The query is returned from
//combination.jsp if use choose
//combination search

query = URLEncoder.encode(query, "UTF-
8");

URL url = new
URL("http://ajax.googleapis.com/ajax/se
rvices/search/web?start=0&rsz=large&v=1
.0&q=" + query);
// opening connection
URLConnection connection =
url.openConnection();
```

The code snippet for parsing the response from Google and storing into array , getting the best URLs, Counting number of results etc using JSON is as follows

```
// Reading the JSON response

   String line;
   StringBuilder res = new
StringBuilder();
   BufferedReader reader = new
BufferedReader(new
InputStreamReader(connection.getInputSt
ream()));
   while((line = reader.readLine()) !=
null) {
    res.append(line);
      //out.println("\n"+line);
   }

//parsing using JSON Objects
 String resp = builder.toString();

  JSONObject json = new
JSONObject(builder.toString());

// Displaying the required results

out.println("Results Ordered = "
+json.getJSONObject("responseData").get
JSONObject("cursor").getString("estimat
edResultCount"));
out.println("<br>");

// Parsing the respone using JSONArray
   JSONArray ja =
json.getJSONObject("responseData").getJ
SONArray("results");

out.println("\n Results:");
out.println("<br>");
for (int i = 0; i < ja.length(); i++) {
out.print((i+1) + ". ");
    JSONObject j = ja.getJSONObject(i);

out.println(j.getString("titleNoFormatt
ing"));
out.println("<br>");
out.println(j.getString("url"));
out.println("<br>");
   }
//out.println(""+response1);
```

4.4 Saving Results and Offline Search

After displaying the results the user has the option to save the results in bundled storage of results. The results can be stored by selecting the checkboxes displayed as prefix to each result. After clicking the checkbox the user can press the save button to save the results (as shown in figure 7).The program search.jsp saves the URLs, search string in the bundle storage. The offline search for the user is taken care by offline.jsp program file. The user can choose offline search from the index.jsp file.

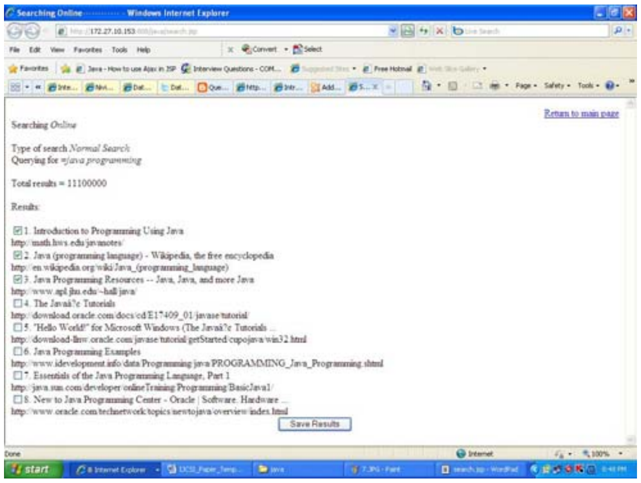
5. Evaluation

This section focuses on evaluating the search performance of this system when chosen online and offline. This section also focuses on the performance of both normal search and combination search. For simplicity, a common search query “java programming” is used.

5.1 Online Search & Saving Results

5.1.1 Normal Search

As shown in the figure 4, after giving the query as “java programming” in index.jsp and clicking “submit query” button google.com returned around 11100000 results as shown in figure 7. For convenience purpose the system


Searching Online .... - Windows Internet Explorer
http://172.27.10.153
File Edit View Favorites Tools Help
Return to main page
Save Results
start

Searching *Online*

Type of search *Normal Search*
Querying for =*java programming*

Total results = 11100000

Results:

[✓] 1. Introduction to Programming Using Java
http://math.hws.edu/javanotes/
[✓] 2. Java (programming language) - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Java_(programming_language)
[✓] 3. Java Programming Resources -- Java, Java, and more Java
http://www.apl.jhu.edu/~hall/java/
☐ 4. The Javaâ?¢ Tutorials
http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/
☐ 5. "Hello World!" for Microsoft Windows (The Javaâ?¢ Tutorials ...
http://download-llnw.oracle.com/javase/tutorial/getStarted/cupojava/win32.html
☐ 6. Java Programming Examples
http://www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml
☐ 7. Essentials of the Java Programming Language, Part 1
http://java.sun.com/developer/onlineTraining/Programming/BasicJava1/
☐ 8. New to Java Programming Center - Oracle | Software. Hardware ...
http://www.oracle.com/technetwork/topics/newtojava/overview/index.html

Figure 7: Displaying results after parsing for search query “java programming”

was restricted to display the first 8 results. The total results are computed using JSON as discussed in earlier section. The next part of the display shows check boxes as a prefix to every result so that user has option to save the result by clicking the checkboxes and later clicking save button.

5.1.2 Combination Search

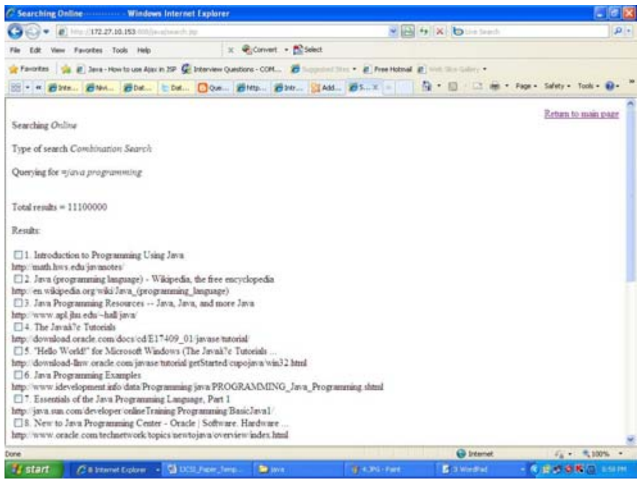
In combination search, the query “java programming” is passed through a function written in combination.jsp program file which give all unique combinations of it.In this case it will also produce “programming java”. The search result of “java programming” and “programming java” yielded different sets of results as shown in figure 8 and 9. The “java programming” query yielded The

11100000 results whereas “programming java” yielded around 12100000 results, which is big difference. Thus this system shows a unique way of getting best search results by giving combination of words and providing user a best possible way to search the internet and choosing the best results. Apart of that, the system also allows to save the results which are yielded from both the search queries, thus providing a variety of options and results.

5.1.3 Offline Search

In searching based on offline, the term “java programming” is submitted to the bundle of stored results. The bundle of results is the one which were saved by the user during normal search as shown in figure 7. The result of this search is shown in figure 10.Figure 10 show only 3 results displayed which are top 3 results stored by the user in figure 7.

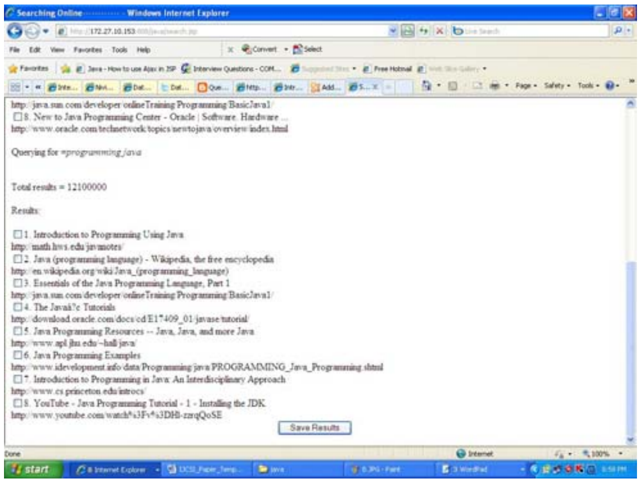
As illustrated in the figures 8 and 9 the combinational search yields different results as compared to the normal search.



```
Searching Online

Type of search Combination Search

Querying for =java programming


Total results = 11100000

Results:

☐1. Introduction to Programming Using Java
http://math.hws.edu/javanotes/
☐2. Java (programming language) - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Java_(programming_language)
☐3. Java Programming Resources -- Java, Java, and more Java
http://www.apl.jhu.edu/~hall/java/
☐4. The Java?e Tutorials
http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/
☐5. "Hello World!" for Microsoft Windows (The Java?e Tutorials ...
http://download-llnw.oracle.com/javase/tutorial/getStarted/cupojava/win32.html
☐6. Java Programming Examples
http://www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml
☐7. Essentials of the Java Programming Language, Part 1
http://java.sun.com/developer/onlineTraining/Programming/BasicJava1/
☐8. New to Java Programming Center - Oracle | Software. Hardware ...
http://www.oracle.com/technetwork/topics/newtojava/overview/index.html
```

Figure 8: Combination Search (part 1), displaying results for search query “java programming”

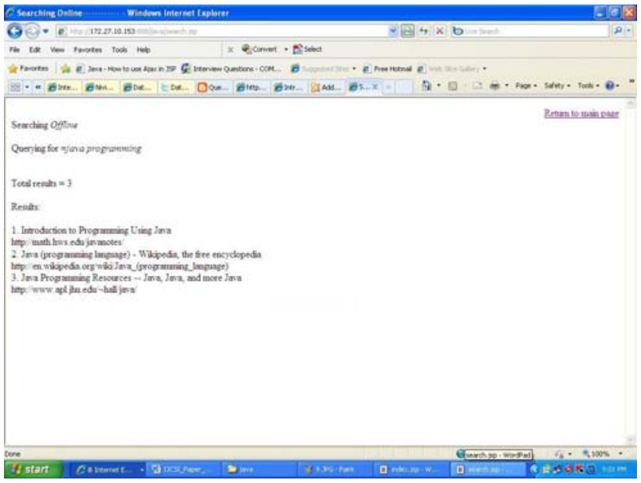
As illustrated in figure:10 the user chooses the option for the offline search and the results are from stored bundle of results. The user after performing combination search, which yields variety of results from combination of words in the query string, can save the best possible results.



```
http://java.sun.com/developer/onlineTraining/Programming/BasicJava1/
☐8. New to Java Programming Center - Oracle | Software. Hardware ...
http://www.oracle.com/technetwork/topics/newtojava/overview/index.html

Querying for =programming java


Total results = 12100000

Results:

☐1. Introduction to Programming Using Java
http://math.hws.edu/javanotes/
☐2. Java (programming language) - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Java_(programming_language)
☐3. Essentials of the Java Programming Language, Part 1
http://java.sun.com/developer/onlineTraining/Programming/BasicJava1/
☐4. The Java?e Tutorials
http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/
☐5. Java Programming Resources -- Java, Java, and more Java
http://www.apl.jhu.edu/~hall/java/
☐6. Java Programming Examples
http://www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml
☐7. Introduction to Programming in Java: An Interdisciplinary Approach
http://www.cs.princeton.edu/introcs/
☐8. YouTube - Java Programming Tutorial - 1 - Installing the JDK
http://www.youtube.com/watch?v=3Fv%3DHl-zzrqQoSE
```

Figure 9: Combination Search (part 2), displaying results for search query “programming java”

Later the offline search yields the best results and thus reduces the repeat searching by the user which again reduces the burden on the search engine server.



```
Searching Offline

Querying for =java programming


Total results = 3

Results:

1. Introduction to Programming Using Java
http://math.hws.edu/javanotes/
2. Java (programming language) - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Java_(programming_language)
3. Java Programming Resources -- Java, Java, and more Java
http://www.apl.jhu.edu/~hall/java/
```

Figure 10: Offline Search, Displaying results for search query “programming java

6. Conclusion and Future Work

This paper proposed an interactive web interface for guided searching and uses powerful Google AJAX search API and JSON to parse the response. This interface provides an option for online and offline search. The proposed system also provides an option for Combination search and Normal search to user. A detailed evaluation of the system demonstrated how the user can harness the capability of search engines by manipulation and automation the existing search engine functions (this paper uses google.com as an example).

However the system has lot of scope to be improved. For instance the system can provide an option to choose between the top search engines like (yahoo.com, window live) thus making the user to see different set of results (this paper uses google.com as example). The system can also be further developed to allow the user to give ranking to the results which are to be saved in the bundle of saved result..

Offline search has lot of scope to improve. A simple replacement policy like least recently used can be employed to update the bundle of saved results and thus making the offline search more efficient and likely one to retrieve best results.

References

[1] Xuehua Shen, Bin Tan, ChengXiang Zhai,"Implicit User Modeling for Personalized Search" *CIKM'05,* October 31–November 5, 2005, Bremen, Germany.

[2] Orland Hoeber and Xue Dong Yang, "Interactive Web Information Retrieval Using WordBars", Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence

[3] M. Schwartz, A. Emtage, B. M e, and B. Neumann, "A Comparison of Internet Resource Discovery Approaches," *Computer Systems,*

[4] G. Nunberg. As google goes, so goes the nation. *New York Times*, May 2003.

[5] Ding Choon Hoong and Rajkumar Buyya,"Guided Google: A Meta Search Engine and its Implementation using the Google Distributed Web Services

[6] The Technology behind Google http://searchenginewatch.com/searchday/02/sd0812-googletech.html

[7] http://code.google.com/apis/ajaxsearch/

[8] Google Groups (web-apis) http://groups.google.com/groups?group=google.public.web-apis

[9] www.json.org

[10] Guidebeam - http://www.guidebeam.com/aboutus.html

[11] Peter Bruza and Bernd van Linder, Preferential Models of Query by Navigation. Chapter 4 in Information Retrieval: Uncertainty & Logics, The Kluwer International Series on Information Retrieval. Kluwer Academic Publishers, 1999.http://www.guidebeam.com/preflogic.pdf

[12] Softnik Technologies, Google API Search Toolhttp:// www.searchenginelab.com/common/products/gapis/docs/

[13] Google API Proximity Search (GAPS) - http://www.staggernation.com/gaps/readme.html

[14] Sree Harsha Totakura, S Venkata Nagendra Rohit Indukuri V Vijayasherly, COS: A Frame Work for Clustered off-line Search

[15] Luiz André Barroso, Jeffrey Dean, and Urs Hölzle, Web Search for a Planet: The Google Cluster Architecture, Google, 2003.

[16] Mitch Wagner, Google Bets The Farm On Linux, June 2000, - http://www.internetwk.com/lead/lead060100.htm

[17] http://json-schema.org

[18] M. B. Rosson and J. M. Carroll. Usability Engineering: scenario-based development of human-computer interaction. Morgan Kaufmann, 2002.

[19] I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.

[20] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41(4), 1990.

[21] B. Shneiderman. Designing the User Interface. Addison-Wesley, 1998.

[22] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large web search engine query log. SIGIR Forum, 33(1), 1999.

[23] A. Spink, D. Wolfram, B. J. Jansen, and T. Saracevic. Searching the web: the public and their queries. Journal of the American Society for Information Science and Technology, 52(3), 2001.

[24] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile construction without any effort from users. In Proceedings of the 2004 World Wide Web Conference (WWW2004), 2004.

[25] J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 2005.

[26] E. M. Voorhees. Query expansion using lexical-semantic relations. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 1994.

[27] C. Ware. Information Visualization: Perception for Design. Morgan Kaufmann, 2004.

[28] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems, 18(1), 2000.

[29] Yahoo. Yahoo search web searvices. http://developer.yahoo.com/search.

[30] Y. Yao. Information retrieval support systems. In Proceedings of the 2002 IEEE World Congress on Computational Intelligence, 2002.

Kalyan Nettii, born in Andhra Pradesh, India. He obtained his Master degree in Technology in Computer Science and Engineering with specialization in Database Management Systems from JNTU, Andhra Pradesh, India, in 2004.Kalyan Netti is interested in the following topics: semantic web technologies, Ontologies, data interoperability, semantic heterogeneity, relational database systems, temporal databases and temporal data modeling.