Another Face of Search Engine: Web Search API's*

Harshit Kumar[1] and Sanggil Kang[2]

[1] DERI, National University of Ireland, Galway
harshitkumar@suwon.ac.kr
[2] Computer Science and Information Engineering, Inha University, 253 Younhyun-dong,
Nam-gu, Incheon, South Korea
sgkang@inha.ac.kr

Abstract. Since search engine development requires building an index which is a tedious and space consuming process. Now-a-days all major search engines provide developers access to their resources through a set of APIs. In this paper we try to answer the following questions. What differences exist between search engines and their associated search APIs? Does search APIs really surrogate for the actual search engine within the research domain? If yes, then which APIs is more suitable? For our experiments, we have used the following search engines and their web search APIs: Yahoo, Google, MSN, and Naver. Our experimental results will help researchers to choose appropriate web search APIs that suit their requirements.

Keywords: Search Engine, Search API, Google, MSN, Naver, Yahoo.

1 Introduction

Lately researchers have been using web interface of search engine [11] or search engine query logs [3] for their research. The basic trouble with these two approaches is that they are manual and tedious. In the year 2002, Google introduced a free SOAP-based search API (Application Programming Interface) [1] that allows public to access their index and further augment it to suit their requirements. Later Yahoo, Naver, and MSN also allowed free access to their index through the public interface of APIs. Researchers now make use of the web search APIs for the collection of data for their experiments. Most of the research works have used Google search API [12, 13] and very few have used either Yahoo [8, 12] or MSN search API [9, 12]. In this paper, we have made an attempt to provide collectively the differences among search results from web interface of search engine and search APIs. We report the following differences: freshness, accuracy, ranking, and the number of results and difference of index.

We have used the search APIs of Google, Yahoo, MSN, and Naver. These APIs have their own limitations regarding the number of queries per day that can be issued,

* This research was partly supported by the Ministry of Information and Communication, Korea, under the College Information Technology Research Center Support Program, grant number IITA-2006-C1090-0603-0031 and by the K-net project under the EU Seventh Frame Programme, grant number FP7-215584.

N.T. Nguyen et al. (Eds.): IEA/AIE 2008, LNAI 5027, pp. 311–320, 2008.

the number of results that they return, etc. Moreover, the search engine does not provide any support for these APIs and most of the information is covert, which leaves users of these APIs wondering about how the APIs interact with the server of search engine?, which index do they access? The results in this paper address the above questions. For evaluation, we have compared the results between two faces of search engine using Discounted Cumulative Gain [2] (DCG), Cumulative Gain [2] (CG), Average Rank [3] (AR), and Kendall-Tau distance [4].

The paper is organized as follows. Next, we discuss the related work which is followed by the evaluation metrics in Section 3. Section 4 has a discussion about experimental results along with analysis. We conclude in Section 5.

2 Related Work

A recent work [5] compares web search results from WUI (Web Search Interface) and web search APIs. It compares how consistent is the index for Google, Yahoo, and MSN APIs with the WUI. We, on the other hand, have a different goal; moreover our evaluation metrics are also different. They claim that their work is the first analysis of the results produced by the Google, MSN, and Yahoo API and WUI interfaces. We also vouch for their claim.

We have found many studies that have used Google API [8], Yahoo API [12] for their research work, and MSN API [9]. There exists one study [10] that compared Google API with Google Search engine. They have shown that the search results from two different Google interfaces: Google API and Google search engine vary in range, structure, and availability. They fail to provide any measurement for the above quantities. We have tried to express the above quantities in number and shown that how much they vary in range, structure, and availability? Most of the studies just compare the results of their proposed method to the search engine using the search engine API. This is the only domain in which search engine API has been exploited. In a recent work [12], the author has listed the issues to be considered while submitting a query to an API. He cautioned that API occasionally fails to return a valid result for a user query, special characters in user query are not supported by APIs. And the number of results beyond the maximum value (2147483647) fails to return a valid response.

Some of the studies [11] copy the results from search engine for the evaluation purpose which is a tedious job. Our work automatically observes user behavior for calculating evaluation metrics. i.e. we store the results returned by the API, user clicked URLs which is used for calculating the values of DCG, CG, AR, and Kendall-Tau distance.

3 Evaluation Metrics

The metric Average Rank (AR) is used for measuring the quality of personalized search. The AR of a query q is defined as shown in Eq. (1).

$$AR_q = 1/|V| \sum_{p \in V} R(p) \tag{1}$$

where $R(p)$ is the rank of URL p, and V is the set of URLs clicked by the user. Second metric that we used for measuring the quality of results is Cumulative Gain (CG). A higher value of Gain Vector symbolizes more relevant results and vice versa. For example, if the highest value of CG is 20 in scenario1 and 12 in scenario 2, that implies scenario1 has high relevant results compared to scenario 2. The CG Vector is calculated as shown in Eq. (3).

$$CG = \begin{cases} G(1) & \text{if } i = 1 \\ CG(i-1) + G(i) & \text{otherwise} \end{cases} \tag{3}$$

Third metric used for measuring the ranking quality is Discounted Cumulative Gain (DCG) as seen in Eq. (4). DCG is particularly useful when results are evaluated at different relevance levels (highly relevant, relevant, and not relevant) by assigning them different gain values. The idea behind DCG is, the greater the rank, the less accessible that URL is.

$$DCG = \begin{cases} G(1) & \text{if } i = 1 \\ DCG(i-1) + G(i)/\log_b(i) & \text{otherwise} \end{cases} \tag{4}$$

For the purpose of this experiment, we have used 3 different relevance levels; $G(i)$=2 for highly relevant result and $G(i)$=1 for relevant result and $G(i)$=0 for not relevant result. Also b is the parameter of penalization; we have taken value 2 for b. To measure the similarity between two lists, we have used a variant of Kendall-Tau distance. Kendall-Tau in its original form cannot be used to compare the results of two search engines because it compares two lists over the same elements. The modified Kendall-Tau that suits our requirements is presented below

$$K^{(p)}(\tau_1,\tau_2) = \sum_{\{i,j\}\in P(\tau_1,\tau_2)} \bar{K}^{(p)}_{i,j}(\tau_1,\tau_2) \tag{5}$$

where τ_1 and τ_2 are the top k lists, and P is the union of all unordered pairs of τ_1 and τ_2. The optimal approach chooses p=0. The value of $K_{(i,j)}$ (τ_1, τ_2) is calculated by formulating the following four cases:

Case I: i and j appear in both top k lists.
(a) i being ahead of j, then the value $K_{(i,j)}$ $(\tau_1, \tau_2)$is 0.
(b) In one list, i is ahead of j and in another list j is ahead of i, the value of $K_{(i,j)}$ $(\tau_1, \tau_2)$is 1.
Case II: i and j both appear in one list and either of them appears in other top k list.
(a) if i is ahead of j in one top k list, then the value $K_{(i,j)}$ $(\tau_1, \tau_2)$is 1 else 0.
Case III: i appear in one list and j appears in another list.
(a) The value of $K_{(i,j)}$ $(\tau_1, \tau_2)$is 1.
Case IV: i and j both appear in one list.
(a) The value of $K_{(i,j)}$ (τ_1, τ_2) is 1.
So as to compare the results over a range, we normalized the value $K_{(i,j)}$ using Eq. (6).

$$K = 1 - (K^{(0)}(\tau_1,\tau_2)/k^2) \tag{6}$$

4 Experiment

Before explaining the experiments, it is important to explain the test collection that was used. An experiment evaluation was carried out using NPL [6] test collection which is available for free from Glasgow IR resources. The NPL test collection is a collection of 93 queries with their descriptions. Fifteen people from different labs at Inha University and Suwon University participated as relevance assessors. They were free to choose any of the 39 queries out of the available 93 queries. However, they have to construct the query term from the given query descriptions. The queries were issued to 8 search engines (4 web search engines and 4 search APIs based search engine). We observed different results for the same query. It is just because different users had different notions for a query description and they inputted different query terms. The experiments were carried out for a period of 3 months from Mid January 2007 to end of March 2007. The total number of queries issued was 450, out of which 80% had the same query description but different query terms were formulated by the user.

The section below gives an introduction to web search APIs and precisely define what we intend to present in this study which is followed by the experiment results.

4.1 Web Search API

In this section, we briefly explain about the search APIs that we used for our work. We developed an interface using Java technologies, HTML Tidy [7], DOM API, and Apache to carry out the experiment. The interface is named Exclusively Your's. It receives a query from the user, forwards the query to the search engine server, and then receives a result set. The result set comprises of URLs, snippets, and title. The result set is then rendered on the browser. Everywhere in the paper result set means a set of URLs. Note that the result set is different from that returned by the web interface of search engine. This can be due to several reasons which we wish to explore in this study. Individual user information such as query issued, results returned (snippets, title), total number of results, and web pages clicked by user were logged in the MS access database which were analyzed later for experiments.

Table 1 presents the list of top 10 titles returned for a query "web search API" requested to web search engine and exclusively yours. Note that both interfaces return the same set of URLs but their rankings are different. For instance, the title "Google Web API" (position 3) has higher ranking in the results from Google search engine whereas the title "Yahoo * Services[1]" has the same rank (position 1, 2) in both interfaces. And the title "MSN Live search API" (position 9) has higher ranking in Exclusively Your's. One thing is sure that the index used by Google search engine and search APIs is not same. Now the obvious question arises, which API is better among Yahoo, Google, MSN, and Naver and which search API results are close to its associated search engine results? We try to answer the questions in the following section.

[1] Yahoo * Services means the sequence of strings enclosed inside Yahoo and Service. In this case, it is Yahoo Search Documentation for Yahoo Web Services and Yahoo Search Web Services.

Table 1. Thr representation of URL titles returned by Google search engine and exclusively yours (Google) for query "web search API"

Position	Google search engine	Exclusively yours (Google)
1	Web Search Documentation for Yahoo! Search Web Services	**Web Search** Documentation for Yahoo! **Search Web** Services
2	Yahoo! Search Web Services	Yahoo! **Search Web** Services
3	Google Web APIs	Alexa **Web Search** Platform
4	Google AJAX Search API - Google Code	Build a Site **Search** with Yahoo! **Search Web** Services
5	Alexa Web Search Platform	Google AJAX **Search API** - Google Code
6	Build a Site Search with Yahoo! Search Web Services	Unofficial Guide to Ask's **Web Search API** - Antezeta
7	» Unofficial Guide to Ask's Web Search API - Antezeta	Decrypting Ask's **Web Search API** - Antezeta
8	» Decrypting Ask's Web Search API - Antezeta	Alexa **Web Search API** - Programmable-Web Profile
9	Alexa Web Search API - Programmable-Web Profile	Live **Search API**, Version 1.1
10	Live Search API, Version 1.1	Windows Live Dev

Table 2. Comparison of Google, Yahoo, Msn and ,Naver web search APIs. Here NL means no limit, NS means no similalrity (50% similar), MS means mostly similar (80% similar), MS* means mostly similar (>90%), and M means many language support.

Comparison Parameter	SE API (G)	SE API (Y)	SE API (MSN)	SE API(N)
No. of query term	NL	NL	NL	NL
No. of results	NL	NL	NL	100
Result Format	Set of URL	Set of URL	Set of URL	XML file
Link:URL	Fewer links	Lots of Links	Moderate Links	NA
No. of queries/day	1000	5000	10000	NL
Result Similarity	MS	MS*	NS	MS*
Development Language	M	M	Microsoft Technologies	M
Repetition of Results	Y	Y	Y	N

4.2 Results and Discussion

In the first set of results, we compare among the search APIs and try to answer the first question i.e. which search API is better and easier to use. In the second set, we compare search API with its associated search engine and report the freshness of results and quality of results using DCG, CG, Average Rank, and Kendall Tau evaluation metrics.

Table 2 lists the comparison of search APIs across various comparison parameters which are listed in the 1st column. The first row compares search engine (SE) APIs for the number of query terms that they can receive. For all the four SE APIs, there is no limit (NL) on the number of query terms i.e. user can enter as many terms as the user wants. The number of results returned by Naver SE API is restricted to 100 and all

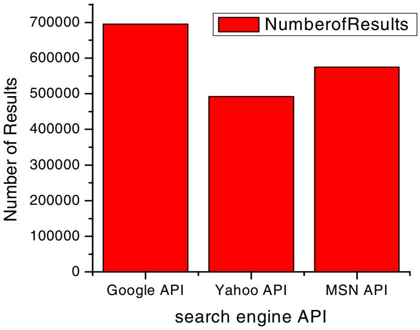


Fig. 1. Number of Results returned by search engine APIs

other SE APIs return infinite number of results. However, the number of results beyond the maximum value (2147483647) fails to return a valid response.

From the graph in Fig. 1, we can see that Google SE API returns on an average more number of results than any other API. This graph is an average of total number of results over all the queries issued by our assessors. This suggests that Google API uses a bigger index among all the APIs which is exactly not true. The explanation follows in the next paragraph.

We were interested to know the number of repetitions in the results from Google API. So, we observed that Google API returns a large number of results but this comes with a lot of repetitions. For instance, the query “AUSDM 2006” issued to Google search API which we issued in March 2007 returned 230 results. The first 100 results were unique with few repetitions. The results from 101 to 110 were exactly similar to results from 91 to 100. The same phenomenon carried on until 221 to 230. Now we wanted to know if Google also shows the same behavior. To verify our doubt, we requested the same query “AUSDM 2006” to Google search engine which returned 346 results out of which 123 were unique. This indicates that Google search engine returns large number of results on the face but actually it is not the real scenario. We also noticed one peculiar behavior: if we issue the same query again, then 20% and 13% of the queries receive different number of results for Google and Yahoo API, respectively. All the APIs return a set of URL, snippet, and title for a user requested query. Whereas, Naver API returns a XML file which imposes an extra burden on the developer to parse the file and extract title, URL, and snippet. There is of course the advantage that the programmer is free to choose any language for parsing XML file. Google and Yahoo search APIs can be used in almost all the languages, on the other hand MSN API only works with Microsoft technologies. The developers who are not comfortable with MSN API will prefer not to choose MSN API. Sometimes the XML file returned by Naver API is not properly formatted which causes many XML parsing APIs to reject the file as not a proper XML compatible version. This is a serious problem with Naver API. To workaround this problem, we avoided using XML parser API and opened XML file as a stream of bytes. We stored the stream of bytes in a string and then parsed the string using string related functions for extracting URL, snippets, title, etc.

The query link: URL returns the number of in-inks that point to a URL. For example, the query "link: www.inha.ac.kr" will return the list of URLs that has an outgoing link to www.inha.ac.kr or in other words it returns the number of in-links to www.inha.ac.kr. Naver API does not support this type of query and Yahoo API is the most supportive i.e. it returns the maximum number of in-links. MSN API and Google API provide a list of few incoming links. There is a limit on the number of the queries that can be requested in one day. Google, Yahoo and MSN API impose a limit of 1000, 5000 and 10000 respectively. Naver API does not impose any such limit.

4.3 Freshness of Results Comparison between API and Search Engine

This section evaluates the similarity between the results from the search API and search engine. Fig. 2 presents the Kendall-tau distance between the two lists: one list comprises top 20 URLs returned from the search API and the other list comprises top 20 URLs returned from the associated search engine. We can see that the Naver API results are almost similar to Naver search engine and that MSN API shows a lot of dissimilarity. The plots in Fig 5 (a), (b), (c), and (d) clearly ascertain the previous statement. If we rank the four web search APIs on the basis of similarity of result set, Naver will be a clear winner followed by Yahoo, Google, and MSN API at the last. The results in Table 2 also support the above statement. In other words, it shows that Naver API returns the freshest results almost same as returned by the search engine. However, there is a caveat here. If we issue the same query again, then 13% of the queries return different number of results for Yahoo API.

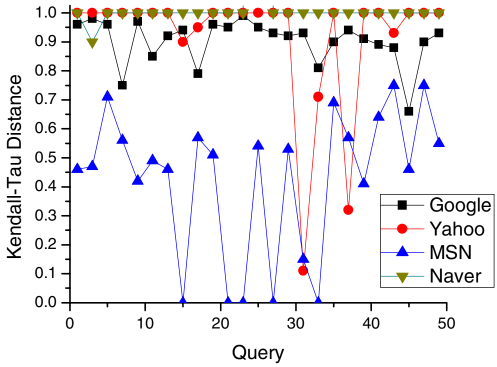
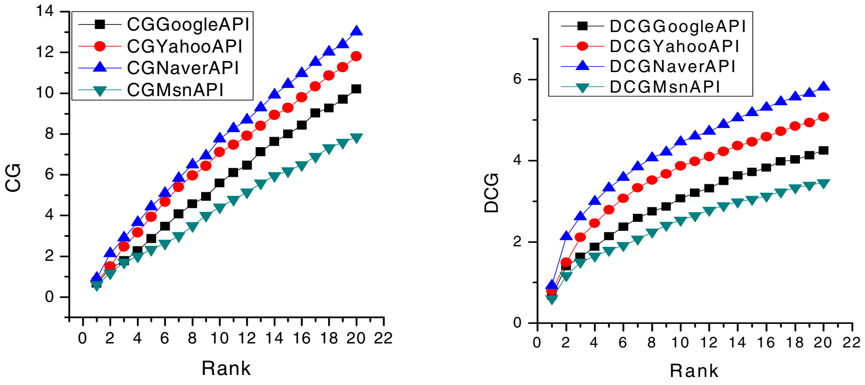


Fig. 2. Kendall-Tau Plot for 50 queries issued to search engine and their associated APIs

4.4 Quality of Results Comparison between APIs

Quality of Results measures how much relevant the returned URL is? To measure the quality, we have used evaluation measures CG, DCG, and AR. AR measures quality on a binary relevance whereas CG and DCG measure the relevance of returned result at more than two relevance levels. Moreover, AR can only be used to compare a search engine with its API but not with another API or with another search engine. Note that in this study we do not intend to compare search engines.



(a) (b)

Fig. 3. (a) Rank vs. CG for search API, (b) Rank vs. DCG for search API

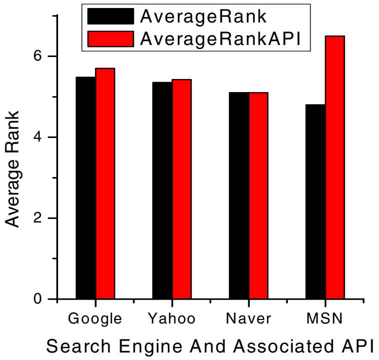


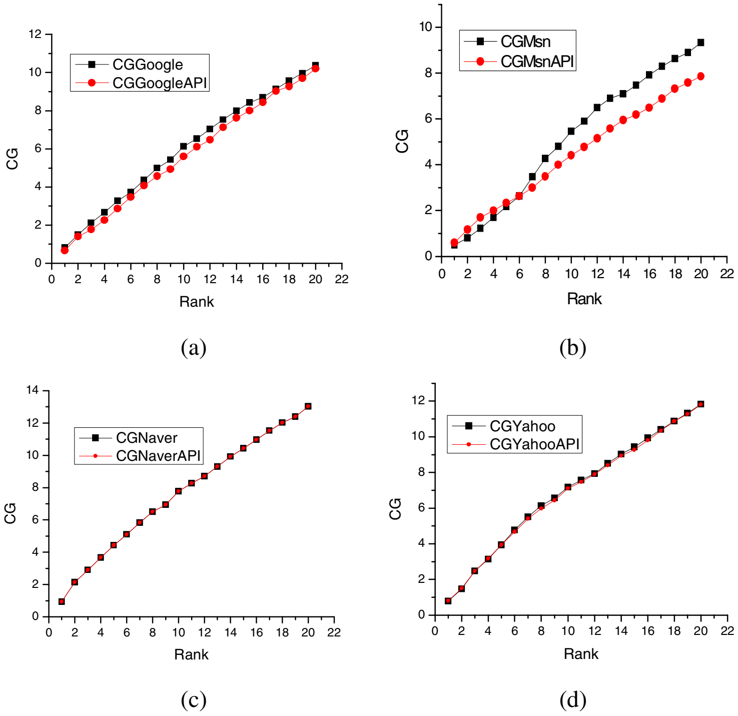
Fig. 4. AR plot for search engine and associated search API

The CG (DCG) plot in Fig, 3 (a) and (b) is plotted as an average CG (DCG) over all the queries. It appears that the Naver API returns high quality of results which is followed by Yahoo API. Note that there is not much difference in the first 10 results from Naver and Yahoo API. The difference increases as assessor evaluates the higher ranked results. The explanation is that the first 10 results might be same or different but assessors evaluated them at almost the same relevance level. However, Naver API returns better results in the 11-20 rank and hence it's CG or DCG value exceeds the Yahoo CG or DCG value. Also the MSN API returns better result in the beginning but later Google APIs takes over. In this case, it is only the first 2 or 3 results which were better but later results were not better than Google.

The comparison of AR between search engine and its API is plotted in Fig. 4. It shows that the Google and its API differ by an AR of 4%, Yahoo and its API differ by 1.3%, Naver and its API by 0% and MSN and its API by 35%. The AR values suggest that the Naver and its' API results are exactly same, which means the Naver search API uses the same index for searching as used by the Naver search engine. While, there is a huge difference in AR between search results from MSN and MSN API. This indicates that the MSN API is using an older index or a smaller index. The research work carried out by [5] has shown that the API indexes are not older, but they are probably smaller.

4.5 Quality of Results Comparison between Search Engines and Its APIs

Fig. 5(c) and (d) support Fig. 4(a) and (b) that Yahoo, Yahoo API and Naver, Naver API have the same search quality. However there is a small difference for yahoo API which is not visible because of averaging. Fig. 5(a) presents the plot of CG between Google and Google API. This plot in addition to the plot of Fig. 4(a) gives us a clear picture of differences between top 20 results. The similar plots for MSN, Naver, and Yahoo is shown in Fig. 5(b), 5 (c), and 5 (d), respectively. From Fig. 5(a) and 5(b), it is apparent that the Google and yahoo API use a different index which is not better than the index used by original search engine. As seen in Fig. 5(d), for Yahoo, the first 10 results have same CG value and the later 10 results have a small difference. The results from MSN show a very strange behavior. The MSM API returns more relevant top 10 results where MSM search engine returns highly relevant next 10 (i.e. from 11 to 20) results. The plot in Fig. 5(c) again ascertains that the Naver API and the Naver search engine use the same index.



(a) (b)

(c) (d)

Fig. 5. (a), (b), (c), and (d) show CG-Rank plot between Google, MSN, Naver, and Yahoo

5 Conclusion

In this paper, we investigated how well an API surrogates for the original search engine. The results suggest that, Naver API very closely surrogates Naver search engine. However there are few drawbacks that the Naver API returns only 100 results and the result quality is lowest among all the APIs. This means, if a researcher is interested in more than 100 results or better search result quality, he should not go with Naver API. On the other hand, Yahoo API is very consistent and its results match closely to the Yahoo search engine but not an exact match. It has another advantage that the result quality is the 2nd best after Naver API. MSN API is very restrictive and works with Microsoft technologies only. Moreover there is a 34% difference in AR between the search results from MSN and MSN API.

References

1. http://code.google.com/apis/soapsearch/
2. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. In: TOIS, pp. 422–446 (2002)
3. Dou, Z.: A Large-scale Evaluation and Analysis of Personalized Search Strategies. In: WWW 16, pp. 581–590 (2007)
4. Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. SIAM Journal on Discrete Mathematics, 134–160 (2003)
5. McCown, F., Nelson, M.: Agreeing to Disagree: Search Engines and Their Public Interfaces. In: JCDL 2007, pp. 309–318 (2007)
6. http://ir.dcs.gla.ac.uk/resources/test_collections/npl/
7. http://tidy.sourceforge.net/
8. Diaj, F., Metzler, D.: Improving the estimation of relevance models using large external corpora. In: SIGIR, pp. 154–161 (2006)
9. http://project.carrot2.org/publications/carrot2-dresden-2007.pdf
10. Mayr, P., Tosques, F.: Google Web APIs – An instrument for Webometric Analyses? In: ISSI 2005 (2005), retrieved from http://www.ib.hu-berlin.de/%7Emayr/arbeiten/ISSI2005_Mayr_Toques.pdf
11. Guha, R., McCool, R., Miller, E.: Semantic Search. In: WWW 2003, pp. 700–709 (2003)
12. Ong, J.: The Predictability Problem, PhD Thesis, http://tertius.net.au/~shade/math/doctoral_thesis.pdf
13. Kumar, H., Park, S., Kang, S.: A Personalized URL Re-ranking Methodology using User's Browsing Behavior. In: KES-AMSTA 2008, pp. 212–221 (2008)