Automatic Framework for Semantic Search Engine Evaluation

Vitalie Scurtu
Dipartimento di Ingegneria e Scienza dell'Informazione
Universita degli Studi di Trento
Via Sommarive 14 I-38100 POVO (TN) - Italy
vitalie.scurtu@studenti.unitn.it

Abstract

The new generation of semantic search engines tends to exploit more and deeper semantic content of documents and queries. As the WEB is emerging to Semantic Web, a new evaluation method is needed for improving the current Information Retrieval Systems. The upcoming generation of Information Retrieval Systems are more document content centric and it has been registered many developments in this field. The evaluation of search engines is crucial for its development between versioning, and shows the right direction to be followed for improvement. Evaluating search engines requires considerable human effort, as large collection sizes demand deep judgments. Manually evaluating search engines is expansive and subjective. In practice it is common to evaluate only on small data sets. Evaluation on small data sets does not cover many aspects of a search engine. In last years it has been increased the interest in automatic or semi-automatic systems for evaluation for its low cost. Previous works on search engine evaluation focused more on its effectiveness, reliability and performance. In this work a new approach for automatic evaluation of Semantic Search Engines is proposed. The proposed evaluation framework takes into account rather the content of the document then the link between the query and the judged document. In the dynamic environment as the WEB, the contents of documents are in a continuous change, and the links between query and right document is in a change as well. The proposed framework evaluates the actual content of the documents and is designed to be flexible, robust, and easy to apply to any type of search engine, independent of the indexed corpus.

1 Introduction

As the amount of available digital information (music, videos, pictures, documents) grows, the performance of Search Engine in accesing these information is becoming critical. Search engines gained popularity, and are the primary tools for navigation in the internet. With the amount of available online web services and data, it is compulsory to use a search engine for finding and accessing them. The current generation of search engines are key-word based, and do not fully exploit the content of the documents. In the amount of the information available on the web, it is easy to loose the track of needed information. In Semantic Web, computer understands the document, and knows what user requests. The importance of content of documents is increasing as the new generations of semantic search engines are improving and gaining popularity.

In previous works on Search Engines (SE) evaluation it has been discussed about the lack of agreement on then right document. In [1] it is shown that the disagreement between assessors has an influence on the relative effectiveness of information retrieval systems. According to [2], the disagreement between assessors is higher for queries with informational requests then navigational queries. However, in [3] it has been shown that assessors disagreement does not destabilize the evaluation and that the agreement between assessors increases when the number of experiments is higher. That means, on a small number of experiments it is likely to have a high disagreement between individuals on what is the relevant document for a query.

Methods of performance evaluation such as coverage, recall and precision are not enough [5]. Because of the characteristics of the WEB such as continuous updates, removal and insertion of the content, the standard methods of Information Retrieval Evaluation do not show the actual performance of the SEs. Experiments in the interactive track of TREC have shown that significant differences in mean average precision in a batch evaluation did not correlate with interactive user performance for a small number of topics in the instance recall and question answering tasks [4].

Though, statistical tests are widely used for showing the performance and reliability of the search engines, experiments are too expansive as small number of judgments are not enough to show the performance of SE. There is an interest in research in finding methods for reducing the number of experiments by randomized sampling and statistical power [6]. However, these estimates require strong assumptions and are not always accurate, especially if the data do not follow the specified distributions [6] [7].

In current work we try to approximate the performance of SE in extracting documents based on AOL queries log of 20 million queries extracted from 6 top search engines, with queries from 650.000 users in a period of over three months (1 March 2006 – 31 May 2006) [8].

For a query *"how to send email anonymously"* the top search engines will give as target documents a form to send the mail anonymously with the explanation and documentation. In AOL query log the target pages for the query are web pages with forms for sending anonymous email with its documentation. However, the right document for the current query depends on the user. A technician might look for source code in PHP or other programming language about how to send email anonymously. Some online mail services also allow users to send email anonymously. Any of the current discussed categories of possible retrieved documents are right for the query, and depends on what exactly user is looking. An evaluation system using a query log will penalize search engines for not retrieving those two links, even if the search engine retrieved many documents that are similar with similar functionalities. The similar situation is when we judge the collection of query and target document. In current paper we do not penalize the SE for not retrieving the right link for a judged document, and we can not do that either, as our experimenting SE indexes a highly technical corpus, while SEs from AOL query dataset indexes general open domain corpus, which includes the corpus indexed by our SE.

We evaluate the SE by scoring the search engine ability to extract relevant document for a given query. The relevance of the document is based on its content, not on the ranking based nor human judgments based. We assume that the perfect document for a query is the link provided by AOL query log on which user has clicked at least once. We do not expect our SE to extract the same document for the query, but we expect to extract a similar document. The more different the document selected by experimented SE is from the AOL query log dataset link, the worst is our SE in extracting the relevant document. Although, the relevance is an ambiguous concept [9][10][11], we assume that our query log includes all relevant documents for a query. When experimented SE extracts similar documents as documents judged as relevant, we give credits to our search engine, since it extracted a relevant document.

On a large scale of data this assumption works [3], as our query log is composed of millions of queries and we can extract queries with multiple target documents judged as relevant. The content of the document is more important then its ranking in top SEs, and SEs should not be penalized for not extracting the top pages. The judged document is interpreted as a prototype, an example of a relevant document for a query, based on which we can score our SE engine performance and coverage.

The current work is organized in following way: In the first part, or the current part, a breef introduction to the problem is presented. In the second part we discuss about the architecture of proposed framework with its subparts and submodules. The main discussed topic is about how to adapt the general corpus queries (Such as AOL queries dataset) to closed domain corpus such as technical articles collection, or other closed domain corpus. Sequentially, in 2.1 2.2 2.3 parts we discuss about filtering queries from open domain such as AOL query dataset for our closed domain search engine, which indexes a big collection of technical articles. İn 2.4 we discuss about the applied document's similarity measure and how it is embedded into our evaluation formula. İn the final subpart of section 2, in 2.5 are presented results of experiements, with proposed evaluation formula and the standard precision and recall formulas.

Finally, in section 4 is shown conclusion and discussion about the current approach.

2 Content Oriented Evaluation

For evaluating the semantic search engine, we try to collect as many queries as possible from AOL query log. According to [3], having a big collection of pairs of queries and documents judged as relevant, will decrease the disagreement between assessments.

Selected queries must give a result on the search engine which does not necessary have the same indexed corpus as they were in AOL SEs from query log collection. For filtering the AOL queries dataset and adapting them to our closed domain search engine, we use a two step filtering method.

First, in pre-filtering phase we exclude navigational queries, queries with no document as target, queries for which the target is a search engine and others queries on which does not have an associated document as relevant.

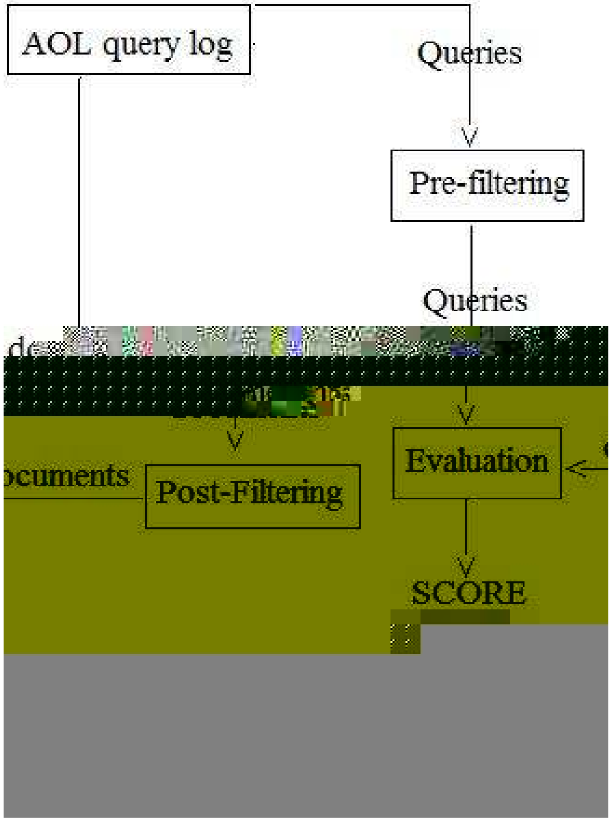
Having the filtered set of queries, we proceed to the second step of filtering, which we also call adapting queries set to indexed corpus by search engine, or post-filtering. We have to ensure that our corpus indexed by search engine contains a relevant document for the given query, and it does not depend on the indexed corpus. Therefore, we have to exclude queries that have as target documents outside our indexed corpus.

The judged documents are considered prototypes, example of relevant document for a given query, and not the most relevant document for the query. It is likely that today for a query used in 2006 the retrieved relevant document to be different. Candidate documents retrieved by evaluated SE can be better or worst then prototype documents in terms of relevance. It is certain that prototype and candidate documents are different, but not totally different, even if they talk about the same topic. Deciding whether a document is relevant for a given query is a human task, and the final application of search engines is to meet human needs.

For approximating the relevance of the document we use similarity measure between candidate documents and prototype (judged) document. In this way we avoid expansive human relevance evaluation, and give a score to the search engine automatically, by exploiting AOL query log. The proposed formula for scoring the search engine is a combination of similarities between judged document and candidate document with its ranking score. The final score of the SE is meant to be scalable and compared with others scores of the same search engine (within versions comparison).

In figure 1 is presented a general view of the system architecture and data flow. We start from the AOL query log and finally, after filtering the queries set we have a collection of queries and documents on which we can evaluate our system.

Figure 1. System Architecture



To mention that evaluated search engine is semantic oriented, therefore, the evaluation based on similarity between prototype and candidate documents. In this way, we focus more on the content of given documents and do not exclude others documents from evaluation, and the search engine gains points when the retrieved documents are more similar to the judged as good document.

Only the top 10 documents given by the search engine are evaluated, since they are on the first page and show how well the search engine retrieves documents. Top N documents for evaluation remains a variable parameter, that can be change and adapted to the needs.

2.1 Filtering queries (Pre-filtering)

In order to create a collection of queries that are semantic oriented, the log of queries is filtered. The quality of filtered queries is crucial for further steps, as in this step we reduce considerable the number of queries for further processing, reducing also the system complexity.

The aim of this step is to remove all the queries except of the informational oriented queries. The first step in filtering the log is to remove queries which are not well formed and queries with no document as target. In AOL log file is present the time stamp of each query, and the number of clicks on the link address. Knowing the time between queries, and the number of clicks, we can assume that a link was relevant for the given query. The very short time between queries is assumed as being an irrelevant document, while links with many clicks and with timestamp between the next query longer, is considered a document judged as relevant.

Navigational queries are the most common queries inputed into SEs, and this is due to the number of services available on the WEB. In AOL query log, around 49% of queries are formed as WEB addresses (such as: subdomain.domain.com). Another category of navigational queries are the names of popular WEB services. Queries like "google" or "yahoo mail" are removed, since they are navigational, meaningless and the expected target will be the links to the very popular web services. In order to identify navigational queries formed as names of services is to check whether queries are included into the WEB address. If it is included in the WEB address, it is likely that the query is formed as the name of the WEB service. For the query "*irs publications*" with the target *"http://www.irs.gov/"*, token *"irs"* is present in both, query and target link address. It is better to have less data, but accurate, instead of many noisy data.

Queries with target document as search engine are removed as well. A list of search engines is created in this scope.

Document itself provides information about its type. Documents with long contents are likely to be informational oriented rather then navigational oriented.

After pre-filtering, the number of queries is reduced to 12% from the original size of queries.

2.2 Running search engine

Having the collection of queries, we can proceed on extracting documents from experimented search engine based on filtered dataset of queries. In previous steps we had filtered our queries, and the number of queries has been reduced to 12% from the original size, that is around 240000 pairs of query and document.

In this part, we input our filtered queries to search engine and extract all the top 10 documents (documents from first page) for the given query. It is likely that our search engine will give few results on queries that are far from the indexed domain. For example, for the query like *"gallstones"*, "*holiday mansion houseboat"* and many others, SE gives to output empty set of documents, because the indexed corpus is closed domain. While for the query like *"lottery"*, *"back to the future"*, it gives an output, though the target document for these two queries are not included in our indexed corpus. For *"lottery"* our SE outputs documents about the software for lotteries, while the judged document from AOL for this query is an online lottery service. Both documents talk about lottery, but they are quit different as one of them is a service, and the other one talks about software solution for lottery services. The query *"back to the future"* is about the popular movie *"Back to the future"* and the judged document from AOL is a database of movies, while our search engine outputs documents containing terms "back", "future", and of course no document about the movie are present in our set. As mentioned before, the query log is for general, an open domain corpus, and the selected queries are still not appropriate for our search engine. However, from the entire query dataset, there is a considerable part of queries with target documents as technical articles. To remind that Web search engines includes our corpus and also other similar corpuses, which will allow us to collect a rich collection of queries with target documents as technical articles. In the next chapter we discuss how we extract queries with target document as technical article from the total amount the queries.

2.3 Selecting queries for closed domain search engine (query Post-filtering)

In this part we classify queries that are appropriate for our corpus from others queries that are not appropriate. That is, queries with target documents included in our corpus from queries with documents that are not included in our corpus. In previous steps we filtered queries and extracted all the documents given by evaluated search engine for those queries. Until here we have a collection of queries with judged documents and another collection of documents which were outputted by evaluated search engine. For clarity reason, the judged documents from AOL dataset are named *"prototype documents"*, and the documents given for the same query by evaluated search engine, which indexes a closed domain corpus, *"candidate documents"*.

Having the prototype documents and candidate documents, we can decide whether the query used for extraction is targeting the closed domain corpus or not, by using similarity measure between candidate and prototype documents.

Therefore, the query is relevant for evaluating our closed domain search engine if the similarity measure between prototype document and at least one of candidate documents is higher then a given threshold.

Similarity measure is widely used in information retrieval, and there are many similarity measures with different purposes. In our experiments we use cosine similarity measure, but it is not excluded to apply different similarity measure such as WordNet semantic similarity measure et al.

Setting a high threshold on similarity between prototype documents and candidate documents will assure us that the search engine contains the target document, or at least a similar document as the target documents, and the query is fair for search engine evaluation, as it ensures that the corpus contains a similar document.

In a dynamic WEB we have modification of documents such as updates, insertion and deletion everyday, and it is almost impossible to decide automatically which document is better then another. Even with human judgments is highly probable to have a high disagreement between assessors. However, on a basic level we can decide if a document is similar to another one. Having a collection judged document, we can consider others similar documents as judged.

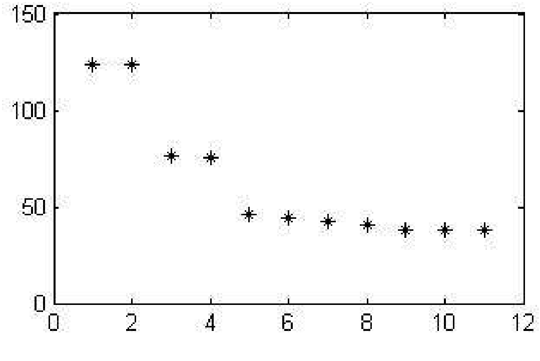
In our experiments we set the threshold for similarity measure as high as 0.8. Candidate documents being less similar to prototype documents does not mean they are less relevant to the query, as it can be more relevant then the prototype document. Setting a lower threshold would bias the collection of query set, as we would allow noisy data in our experiment set. It is better to have less data, but accurate, then more data but noisy.

2.4 Evaluating the search engine

Our method for evaluation requires the inclusion of target documents similarity, and the Tf-Idf score assigned by SE for each document. Tf-Idf is the weighting score widely used by IR systems, and in practice it is commonly used to combine Tf-Idf with some others weights [14]. In any case, Tf-Idf weight is the value based on which documents are ranked.

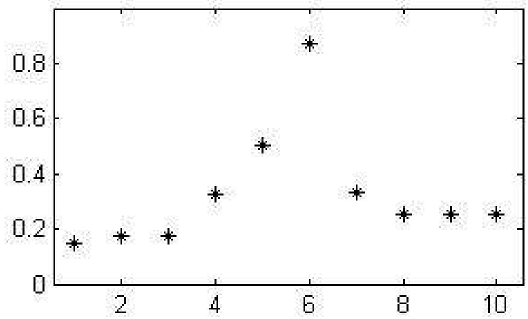
In current chapter we propose a new formula for scoring search engine that takes into consideration the ranking of documents based on Tf-Idf score. The proposed formula aims at covering the gaps in document ranking, and this is due to big differences between Tf-Idf scores of documents. In figure 2 are presented Tf-Idf scores of documents for a query.

Figure 2. Ranking Scores



There are 4 documents with relatively high Tf-Idf scores, and the SE should be graded for giving such high scores to those 4 documents only if at least one is high relevant. In contrary case, it should be penalized. In figure 3 is presented the similarity measure between prototype document (judged), and candidate documents (retrieved by SE on the first page). The document with similarity measure of 0.8 is actually the same document as prototype, and similarity measure is not the same because of the insertion into the actual document (prototype document is updated while SE indexed a previous version of the same document).

Figure 3. Similarity Scores between *"prototype document"* and *"candidate documents"*



The most relevant document for the query is ranked as number 6 with a relatively low Tf-Idf score, equal to 49. In case of an inclusion into corpus of three similar documents as those top 4, the relevant document which is number 6, will not appear on the first page. Therefore, the difference between scores of top 2 ranked documents and the rest, is called a gap, and the rest of the documents are present in the first page only because of the lack of others similar documents as first top 2 documents. By adding only 5 documents that are similar to top 4 documents, we move the most relevant document on the second page (will be ranked on the 11th position).

If the score of the search engine is higher, and the similarity between the pairs of documents is lower, the search engine is retrieving documents that are farer to document judged by user as relevant. On the other hand, the higher the similarity between the documents is, the better the engine performs the search. When the score is lower and the similarity is lower, that means that the search engine predicts that the document is not very relevant for the query, and it performs well. We do not penalize the search engine if it didn't retrieve the more similar document to the query, we penalize when it retrieves a document with a high Tf-Idf score for a document that is far from the selected document as golden standard.

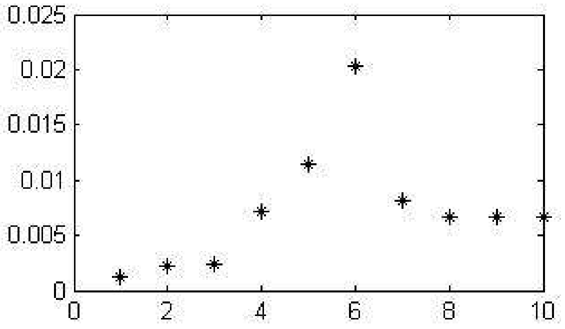
The rate of "how well the search engine performed on a query" is: similarity between documents divided by the Tf-Idf score given by search engine. Therefore, the proposed formula for performance evaluation is:

$$\frac{1}{N}\sum_{k=1}^{N}\frac{\text{similaritiesV(k)}}{\text{scoresV(k)}}$$

Where *similaritiesV* is vector containing similarity measures between candidate documents and prototype document for query 'k', and *scoresV* is vector containing Tf-Idf scores given by search engine for each candidate document for query 'k'. N is the number of pairs of query and associated judged documents. The current formula is applied for scoring the performance of each pair of query and document, and the final SE score is the average of all score.

In figure 4 it is presented the score of each retrieved document for one query, according to scores shown in figure 2 and 3. SE gained poor grades for top for pages, while for the rest documents it graded with relatively higher grades. The maximum score is for the sixth document, and it is 0.0204.

Figure 4. Combined scores of candidate documents



The proposed formula does not depend on the number of documents extracted by search engine, as we give an average score per query. For example, if SE retrieves only one document with a high Tf-Idf score, which is also very similar to prototype document (is the same), SE gains high grades.

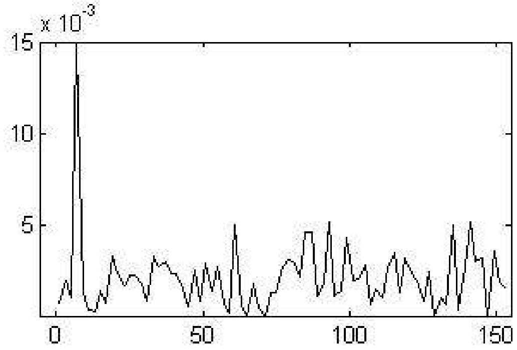
It also aims at solving problem of scoring ambiguous queries, or queries with multiple target documents, as it gives very low grades for non relevant documents extracted in top and higher grades for documents extracted, even if they are in last position. And it gives very high scores to SE only when it extracts documents with high similarity measure between prototype documents with high Tf-Idf score. When SE retrieves all relevant documents for one query, it is graded many times with high scores.

It is also able to predict the performance of search engine in case of inserting new documents into indexed corpus, as it takes the Tf-Idf into account, which is a very precise value about the ranking of documents. Tf-Idf score depends on the search engine, and it is not an absolute value, therefore it can not be compared with others SEs. Tf-Idf is a weight that allows a relative comparison of documents within the same SE.

2.5 Experiments and results

After applying all the described above steps, we collected over 144 queries that has as target technical articles documents, targeting the closed domain corpus indexed by evaluated Semantic Search Engine. In Figure 5 are presented results of proposed formula for each pairs of query and document.

Figure 5. Vector of scores



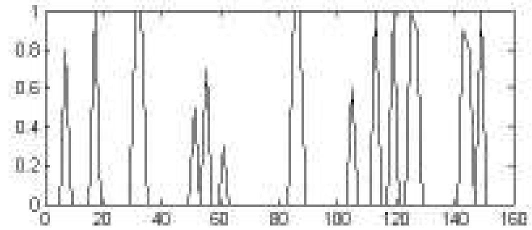
The curve of performance is oscillating between local maximum and minimum, which shows that we have a randomization in our set of queries. A good randomized sampled set of query is reliable and shows the actual performance of the SE. The global maximum point shown in Figure 5 is a case where our search engine performs the best, where it retrieved top 5 documents very similar to the prototype with a high Tf-Idf score, while the rest 5 documents are very far from prototype with very low Tf-Idf score.

Bpref is the precision measure used in practice for evaluating SEs performance [13], and the formula is:

$$\text{bpref} = \frac{1}{R}\sum_{r} 1 - \frac{|n \text{ ranked higher than } r|}{R}$$

Where for a topic with R relevant documents and r is a relevant document and n is a member of the first R judged non-relevant documents as retrieved by the system. In our case, relevant documents are candidate documents for which cosine similarity measure between prototype document is higher then threshold, that is 0.8. In figure 8 are presented pbref for each query, which are later summed and divided by the number of relevant documents.

Figure 8. bref score foat each query.



According to results, pbref of evaluated SE is 0.8017 while recall is: 0.2597. That means, the SE is able to rank very well the documents according to its relevance, while the retrieval performance is very low, since 74.03% of relevant pages are not within top 10 pages (the first page). Pbref value is good for comparing the performance between search engines, as it gives absolute values about precision and recall.

In comparison with precision and recall, current proposed formula is a cumulative performance scoring formula of the search engine, and it is designed to give relative performance score of search engine within versioning, and not a global comparison of search engines. That is, we can check whether the SE improves its performance by comparing with previous results.

3 Conclusion and Discussion

In current report we had proposed new methods of collecting pairs of queries with judged documents from queries log that are targeting to a general, open domain corpus collection of documents. The framework is able to adapt it to a search engine that does not necessary index the same type of corpus. The platform for collecting set of queries is easy to implement, as it uses basic IR techniques. In the same time, once there is available a rich query log, it is easy to apply the methodology described, with the possibility in applying different performance evaluation formulas. The platform is meant to be flexible, robust and easy to implement.

The speed can be also improved in further development, as it is possible to embed the modules, or a redesigning of the platform to have an "on-fly data processing". The sub-modules can be replaced as well. For query classification in current work has been used Naïve classification method, therefore it can be designed a more advanced linear classifier that takes information about queries, documents and search engine as classification features.

It is also proposed a new formula for SE performance evaluation, which scores the search engine by retrieved content, with no need of extra human evaluation, as it assigns scores by documents similarities. Human judgments, however, is needed from the beginning, as it provides example of successful searches. Along with the proposed formula, we also computed precision and recall, based on the query and document extracted from AOL query log.

In my opinion, the collection of 144 pairs of queries and documents are not enough to evaluate the real performance of SE, as there are few examples of ambiguous queries, or queries with multiple target documents. The indexed corpus is a collection of technical articles, and this automatically reduces the ambiguity. It would be a good idea to change the indexed corpus according to the number of queries targeting into the same domain, in order to maximize the number of pairs of queries and documents. To mention that indexed corpus can be changed, and normally it is done automatically.

AOL query dataset is an example of available dataset that can be successfully exploited for research, and in our case, for Search Engine evaluation. The collection is set up in 2006, by that time users had a different concept about SE than it is today. The most important in evaluating SEs is to have a reliable and trustable collection of pairs of queries and judged documents. Unfortunately, there are many constraints in publishing query logs because of privacy and security reasons.

In [15] it has been mentioned about automatic queries generation based on indexed documents. It has been few works done in this direction, and this is due to the difficulty in generating artificially reliable samples of queries, since the queries would then be unrepresentative of real users' needs.

References

[1] S. P. Harter. 1996. *Variations in relevance assessments and the measurement of retrieval effectiveness.* Journal of the American Society for Information Science, 47(1): 37-49.

[2] Voorhees, E. M. 2001. *Evaluation by highly relevant documents.* In ACM Conference on Research and Development in Information Retrieval, pages 74–82, New Orleans, LA.

[3] Voorhees, E. M. 1998. *Variations in relevance judgments and the measurement of retrieval effectiveness.* In ACM Conference on Research and Development in Information Retrieval, pages 315–323, Melbourne, Australia.

[4] Turpin, H., and Hersh, W. 2001. *Why Batch and User Evaluations Do Not Give the Same Results.* In Proceedings of SIGIR'01, ACM Press, pages 225-231, New Orleans, LA

[5] J. Bar-Ilan. 2002. *Methods for measuring search engine performance over time.* Journal of the American Society for Information Science and Technology, 53 (4):308-319.

[6] ERIC C. JENSEN. 2006. *Repeatable Evaluation of Information Retrieval Effectiveness in Dynamic Environments,* PHD Thesis, Chicago, Illinois

[7] Kupper, L. and K. B. Hafner. 1989. *"How appropriate are popular sample size formulas?".* The American Statistician, 43:101–105.

[8] G. Pass, A. Chowdhury, C. Torgeson, *"A Picture of Search"* The First International Conference on Scalable Information Systems, Hong Kong, June, 2006.

[9] Borlund, P. 2003. *The concept of relevance in IR.* Journal of the American Society of Information Science and Technology, 54(10):913 – 925.

[10] Greisdorf, H. 2000 *Relevance: An interdisciplinary and information science perspective.* Informing Science, 3(2).

[11] Mizzaro, S. 1996 *Relevance: the whole (hi)story.* Journal of the American Society of Information Science and Technology, 48(9):810–832.

[12] Resnik P. 1995. "*Using information content to evaluate semantic similarity".* In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal. pages 448-453.

[13] Buckley, C. and E. M. Voorhees. *"Retrieval evaluation with incomplete information."* In ACM Conference on Research and Development in Info rmation Retrieval, Sheffield, UK, 2004.

[14] A. Singhal. *"Modern Information Retrieval: A Brief Overview"*, Google, IEEE Data Eng. Bull, 2001

[15] Buckley, C. *"Proposal to TREC Web Track mailing list"*
http://groups.yahoo.com/group/webir/message/760
November, 2001.