A Case Study in Web Search using TREC Algorithms

Amit Singhal*
AT&T Labs — Research
180 Park Avenue
Florham Park, NJ 07932, USA

Marcin Kaszkiel
AT&T Labs — Research
180 Park Avenue
Florham Park, NJ 07932, USA

ABSTRACT

Web search engines rank potentially relevant pages/sites for a user query. Ranking documents for user queries has also been at the heart of the Text REtrieval Conference (TREC in short) under the label *ad-hoc* retrieval. The TREC community has developed document ranking algorithms that are known to be the best for searching the document collections used in TREC, which are mainly comprised of newswire text. However, the web search community has developed its own methods to rank web pages/sites, many of which use link structure on the web, and are quite different from the algorithms developed at TREC. This study evaluates the performance of a state-of-the-art keyword-based document ranking algorithm (coming out of TREC) on a popular web search task: finding the web page/site of an entity, *e.g.* companies, universities, organizations, individuals, etc. This form of querying is quite prevalent on the web. The results from the TREC algorithms are compared to four commercial web search engines. Results show that for finding the web page/site of an entity, commercial web search engines are notably better than a state-of-the-art TREC algorithm. These results are in sharp contrast to results from several previous studies.

Keywords

Search engines, TREC ad-hoc, keyword-based ranking, link-based ranking

1. INTRODUCTION

With the explosive growth of the World Wide Web, finding useful web pages/sites using web search engines has become a part of our everyday lives. According to a recent study sponsored by RealNames Corporation, 75% of frequent Internet users use search engines to navigate the web [2]. With such usage, search engines continually strive to improve their performance. What search algorithms work best for finding information on the web? This has become a critical question given the heavy use of search these days.

Traditionally search algorithms have been studied in the Information Retrieval (IR) research community [16]. Most traditional algorithms are keyword-based[1] and, given a user query, use word frequencies, word importance, document length and other statistical cues to assign potential importance to a document. However, with the emergence of the web many new algorithms for web search have been proposed and are being used in various web search engines today. Many of these algorithms incorporate link-structure of pages in their ranking schemes, and are notably different from the traditional keyword-based document ranking algorithms.

One conference which has been quite influential in the advancement of traditional keyword-based IR ranking algorithms is Text REtrieval Conference or TREC [24]. TREC is a series of annual conferences run by DARPA and NIST with the aim of objectively evaluating text search and related technologies in independently run evaluations. Valuable benchmark test collections are produced as a by-product of TREC. The document search problem has been dubbed as *ad-hoc* search under TREC. The ad-hoc scenario is parallel to what happens in web search. A user provides the search system with a (usually short) query, and the system ranks potentially relevant documents in response to the query. Traditionally TREC has used documents from Newswires and other non-web text collections, for example, AP Newswire, Wall Street Journal, LA Times, San Jose Mercury News, the Federal Register, etc. More recently there has been a shift towards using a web document collection [7].

During the last eight years, TREC participants have developed new document ranking algorithms that have been shown to be quite effective for searching document collections used in the TREC ad-hoc tasks. One difference between the traditional IR or TREC environment and the web environment is the presence of hyper-links between web documents. Several search techniques have been proposed in the web environment that exploit the presence of links [1, 9]. Major web search engines don't disclose all details about

*This work was performed when the authors were employees of AT&T Labs. Current contact information for both authors: Google, Inc., 2400 Bayshore Pkwy., Mountain View, CA 94043, USA, {singhal, martink}@google.com

[1]We use the term *keyword-based* to refer to algorithms that do not use any linkage information, and the term *link-based* for algorithms that use both keywords and links in their ranking schemes.

WWW10, May 1-5, 2001, Hong Kong.
ACM 1-58113-348-0/01/0005.

their ranking schemes, however, it is widely known that several of them do incorporate link information in some form [1, 25]. How much more effective are link-based methods in the web environment as compared to a state-of-the-art keyword-based method developed for the TREC ad-hoc task? This question has been studied in a limited number of studies, especially under TREC's web track [5, 6, 7]. The results from these studies indicate that for web search, link based methods do not hold any advantage over the state-of-the-art keyword-based methods developed for TREC ad-hoc search. These results are quite counter-intuitive given the general wisdom in the web search community that some kind of linkage analysis does improve web page/site ranking. Our work is motivated by this discrepancy between the results presented in [5, 6, 7], and the general belief in the web search community.

Different web search engines make competing claims regarding their coverage and search effectiveness. In this study, we don't concentrate on comparing the search effectiveness of different web search engines. There have been several studies that do such a comparison [4, 11]. Instead, our aim is to study how a state-of-the-art keyword-based document ranking algorithm (emerging from the TREC ad-hoc task) will perform on a realistic web search task; and how that performance compares to the performance of some popular web search engines which use link structure in their ranking schemes. Previous studies have shown that link-based methods do not hold much advantage over keyword-based TREC ad-hoc algorithms, however, these studies accompany their results with several caveats which we discuss in detail in Section 2. This work aims at studying the above question in an environment which is closer to real web search and does not have these caveats. Again, the details are discussed in Section 2.

The rest of the study is organized as follows. Section 2 discusses the TREC ad-hoc and web tracks and points out some of the shortcomings of the web search evaluation studies done under TREC. Section 3 discusses our experimental environment and explains how this environment removes the problems associated with the previous studies. In Section 4 we describe our implementation of a state-of-the-art TREC ad-hoc algorithm, and show that it indeed is competitive with the best TREC results. In Section 5 we present our results and discuss them. Section 6 concludes the study.

2. THE TREC AD-HOC AND WEB TRACKS

The ad-hoc task has been at the heart of TREC evaluations since the begining of TREC [24]. In this task, TREC participants are given a collection of Newswire and other documents, usually about 500,000 to 700,000 documents in roughly two gigabytes of text. Along with the documents, the participants are also given a set of fifty queries posed by real users (often called *assessors* as their key role is to assess the relevance of documents retrieved by different systems for their queries). The conference participants rank documents from the collection for every query using their systems, and the top 1,000 documents for each query are returned to NIST by every participant for evaluation. The assessors judge the top 100 to 200 documents from every system for relevance and various evaluation scores are computed for each participating system (for example, average precision, precision in top 10, 20, 30 documents, and so on).

Even though the TREC ad-hoc task, especially when using short 2-3 word queries[2], is very close to what happens in a web search system, there are some notable differences. For example, the type of documents being searched in TREC ad-hoc are not web pages.

TREC Web Tracks

An effort to evaluate web search is underway at TREC under the web track [7]. This track deals with some of the differences between web search and TREC ad-hoc, and uses an evaluation framework based on web data. This track uses a collection of web pages from an early 1997 crawl of the web done by the Internet Archive [5]. The queries are either selected from a web search engine's query log [5, 7], or are the queries provided by the NIST assessors for the regular TREC ad-hoc task [6, 7]. The evaluation measure used is precision in top twenty pages retrieved (*i.e.*, proportion of relevant pages in top 20) [5, 6, 7].

One of the main aims of TREC web track has been to answer the question if link-based methods are better than keyword-based methods for web search. Most results coming out of the web track indicate that (as measured under TREC) link-based methods do not have any advantage over a state-of-the-art keyword-based TREC ad-hoc algorithm. For example, according to Hawking et al. in [6]:

> *... results are presented for an effectiveness comparison of six TREC systems ... against five well-known Web search systems .... These (results) suggest that the standard rankings produced by the public web search engines is by no means state-of-the-art.*
>
> *... all five (public web) search engines performed below the median P@20 for (short) title-only (TREC) VLC2 submissions ...*

Also, in [7] Hawking et al. say that

> *... Little benefit was derived from the use of link-based methods for standard TREC measures on the WT2g collection. ... One group investigated the use of PageRank scores and found no benefit on standard TREC measures. ...*

On a similar note, Savoy and Picard in [17] say that as implemented in their study:

> *... Hyper-links do not result in any significant improvement ...*

Overall, the sentiment in [5, 6, 7, 17] is that when applied to web search, state-of-the-art keyword-based techniques used in TREC ad-hoc systems are as effective as link-based methods. Hawking et al. do accompany these counter-intuitive results with several shortcomings of the TREC environment that might be causing them. For example, in [7] they say:

> *... The number of inter-server links within WT2g may have been too small or it may be that link-based methods would have worked better with different types of queries and/or with different types of relevance judgments. ...*

[2]Each query also has much longer versions which some TREC participants use in their systems.

These caveats to the results presented in [5, 6, 7] are the main focus of this study. We observe the following shortcomings of the evaluations done in the TREC web track, and design a new evaluation which is aimed at removing these shortcomings to study the effectiveness of link-based vs. keyword-based search algorithms again:

1. The queries used in the TREC environment are mostly topical, *i.e.*, they are aimed at finding relevant pages on various topics. Whereas in a real web search environment the users pose many different kinds of queries, *e.g.*, find a particular site, or find high quality sites on a topic, or find merchants that sell something cheap, etc.,

2. The relevance judgments used in [5, 6, 7] are done on a per page basis and not on a per site[3] basis. Even though the evaluation measure used—precision at rank 20—rightly measures the precision oriented nature of web search users, the page-based judgments ignore the (site-based) browsing aspect of the web.

   For example, in doing an in-house pilot study, we found that for the query "new york city subway" (posed by one of our users) our TREC ad-hoc algorithm retrieved eighteen out of the top twenty pages from the site `www.nycsubway.org`, and all were judged relevant by our user. Most commercial search engines realize that this is not very desirable from the users' perspective, once on the site `www.nycsubway.org`, users like browsing the pages on that site themselves. Therefore, most commercial search engines group the results by site. Page based precision measurement tends to favor TREC ad-hoc algorithms which can retrieve twenty pages, all relevant, from a single site. On the other hand, site-based grouping done by most commercial web search engines artificially depresses the precision value for these engines (as measured under TREC) because it groups several relevant pages under one item and fills the list of ranks by other, possibly non-relevant, sites.

   The problem that all relevant documents are not *pertinent* to a user is a long standing problem in retrieval evaluation [3]. Since pertinence is hard to quantify, most retrieval evaluations just use document relevance as the evaluation criteria. The web search engines, and in our opinion rightly so, take the view that multiple pages from the same site, even though relevant, are less pertinent as compared to relevant pages from different sites. The TREC evaluations ignore this aspect.

3. The web collection used in TREC evaluations is a 100 gigabyte collection with 18.5 million pages based on an early 1997 crawl done by the Internet Archive [5]. This collection is quite outdated with respect to the link structure of the current web. For example, the average number of cross-host out-links in the TREC collection is 1.56 per page, whereas in a recent crawl of the web we notice that the average number of cross-host out-links is 4.53 per page, almost three times as many. This indicates that there is a lot more linkage to be exploited in the current web compared to the web data used in TREC. This observation holds across all the comparative measurements we did. For example, the average number of in-host out-links is 5.57 per page for the TREC data, but it is much higher—11.63/page—for our recent crawl. Similarly, the average number of cross-host in-links for the TREC data is 0.12/page and this number is 2.08/page for our crawl. (There is a difference in the average number of out-links and the in-links per page because the out-links include links pointing to pages not in out collection.) Also, the average number of in-host in-links per page is 3.71 for TREC data and it is 6.31 for our crawl. (In all these measurements we only count links that have some valid anchor text attached to them.)

4. In a binary relevance model, as used in TREC, there is no notion of a relevant page being more or less relevant than another relevant page. However, on the web, there are clearly good and not so good pages on every topic. The quality of a web page is a subjective issue and the search engines tend to capture it numerically by (for example) the number of outside pages that point to a given page (assuming linkage as a form of recommendation of quality), and other such heuristics. This aspect is not captured in TREC evaluations.

5. Concentrating on particular results presented in [5, 6] which show that the commercial web search engines are notably worse than modern TREC ad-hoc algorithms, we want to emphasize that in [5, 6] the precision results for TREC algorithms are obtained on the TREC web data, whereas these results are compared to commercial web search engines running on a completely different and much larger and recent web crawl. Hawking et al., do acknowledge that this difference (in the underlying web collections) is a potential source of inconsistency in their results.

These shortcomings of previous work do not give us confidence that the results from these studies will hold in a realistic, more recent web search environment. In the following section, we describe our experimental environment which is aimed at removing these shortcomings and evaluating the effectiveness of current link-based web search engines vs. a state-of-the-art keyword-based TREC ad-hoc algorithm.

3. EXPERIMENTAL ENVIRONMENT

To set up an experimental environment that would allow us to objectively study the effectiveness of TREC ad-hoc algorithms for a realistic web search task, we need the following tools:

- **Queries**: A set of real user queries.
- **Judgments**: User judgments for the "goodness" of a page/site retrieved by a search engine.
- **TREC Implementation**: An implementation of a state-of-the-art TREC ad-hoc algorithm.
- **Evaluation Measure**: An objective evaluation measure that accurately measures search effectiveness for the search task at hand.
- **Collection**: A large collection of fresh web pages.

In the following we discuss our approach to selecting and building each of these components.

[3]We loosely use the term *site* to refer to the root page of a group of pages on (usually) the same host. Of course this definition is not always true.

Queries and Judgments

As we mentioned earlier, there is a wide variety of query types that users pose to a web search engine. Each type of query defines a specific web search task. For example, one could easily identify queries in a search engine query log that seek a particular web site, *e.g.*, "Purdue University Homepage", "American Airline web site", "Newark Airport", and so on. These queries can all be grouped under the task: *find a web page/site*. Similarly, there are a lot of queries that seek high quality sites on a certain topic, *e.g.*, "jazz", "search engines", and so on. These queries can be grouped under the task: *find high quality web pages/sites on a topic*. One can identify many such task groups in a search engine query log. Based on our informal analysis of a large query log, the two tasks we mention above: *find a web page/site* and *find high quality web pages/sites on a topic* are quite popular among web users. Short of doing a lot of manual classification, it is hard for us to quote concrete numbers on how prevalent each query type is on the web. To our knowledge, there is no published study that groups queries from a query log into such task groups. Most studies done on search engine query logs study various statistical properties of the queries, *e.g.*, the average query length, query repetition, and so on [8, 18].

To select a set of queries for use in a web search evaluation, ideally, one should take a random sample from queries posed to a search engine by a large population of users. Also, the pages retrieved by different engines should be judged for goodness by the person who posed the query. However, the two goals of a) using a large population of users, and b) asking the original user to do relevance judgments, are quite contradictory in a lab setting. One possible fix to this problem is to use a limited set of users available for an experiment and only use their queries, and their judgments. This approach does not yield as wide a variety of query types as one can get from a real search engine query log. The other fix to this problem is to use a sample of queries from a real search engine query log, and ask a human subject, obviously different from the person who posed the query, to judge pages for someone else's query. This is the approach taken in some of the TREC studies [5, 7]. In essence, the human subject is told: "make your judgments based on what you would have been looking for, had you posed this query". This approach suffers from the problem that two human interpretations of a query can be quite different. For example, a human subject can interpret the query "who wants to be a millionaire" as a query looking for ratings/reviews of the famous TV show, whereas the original user who posed the query might have been looking for the home-page for the show.

Given these problems in obtaining extensive relevance judgments, and given that it is quite time and human-labor intensive to get relevance judgments from humans for a large set of queries, we decided to experiment with only one popular type of web queries for which doing relevance judgments is relatively easy; we use queries of the type: *find a web page/site*. A large proportion of users pose such queries to web search engines everyday, and doing relevance judgments for these queries is not as expensive. This selection also allows us to do our evaluation using a relatively large set of test queries.

From two real user query logs, one our internal log for our engine, and another made available by Excite (`www.excite.com`) we select queries that are explicitly seeking a home-page or a web-site. The Excite log[4] contains 2,477,283 queries posed to Excite during few hours on Dec. 20, 1999. To avoid the query interpretation problem mentioned above, we first find all queries in these logs that contain the string `home` followed by the string `page`, or the string `web` followed by the string `page` or `site`. This strict selection criteria gives us 14,603 queries from this log, for example "Aces High homepage", or "Champion Nutrition web site". There are many more queries in the log that seem to be seeking a web page/site (*e.g.*, "Panache communications" or "Office Depot") but we don't want to get engaged in a query interpretation exercise. Then we use a human subject to go through these 14,603 filtered queries, and a) eliminate the ones that are not seeking an explicit page, *e.g.*, "web site administration", and b) link queries to their respective web pages, *e.g.*, link "Purdue University Homepage" to `www.purdue.edu`. Using this process, we generate a set of 100 queries, and their corresponding relevant pages, for use in our evaluation.

Since the keyword-based TREC algorithms are quite sensitive to presence of extraneous words (like `homepage`) in a query, the human subject generating the <query, relevant page> pairs also removed these extraneous words from the queries. So the query "Champion Nutrition web site" was reduced to just "Champion Nutrition". To our knowledge, most web search engines have such a stop-list (list of words to remove) for query processing. Despite our instructions, eight of the 100 queries were left as is by our human subject and do contain these extraneous words.

Our query selection process eliminates the first, second and fourth problems (mentioned in Section 2) with the previous studies done in [5, 6, 7]. Since we have only one page that is relevant to a query, the fourth problem of differences in quality of two relevant pages does not exist. Also, the larger problem (problem 2 in Section 2) of page-based, instead of site-based, evaluation disappears since there is only one correct site for a query.

We realize that for queries that seek a web site, it is possible for an engine to use some URL based heuristics to improve its chances of finding the relevant site. For example, for the query "IBM", it is a reasonable guess that the user is looking for the site `www.ibm.com`. If the commercial web search engines use such URL based heuristics, they will have an unfair advantage over the TREC algorithms. For this reason, in our query selection process, we take extra care to make sure that the desired site is not a URL formed easily by using query words. For example, we reject queries like "IBM", or "AOL", or the query "williams sonoma homepage" as the desired page (`www.williams-sonoma.com`) has query words in the URL. Even though there is nothing wrong in using such URL cues to rank pages for a query, we want to limit the advantage the commercial engines might have due to using such cues. For the queries used in this study, if the commercial engines do use some URL cues to promote certain pages, they must do some non-trivial processing of the query string to match it to a page URL. One variable that we did not account for in our query selection process was keyword navigation services like RealNames. We discuss the impact of this on our results in Section 5.

[4]Made available by Jack Xu of Excite via *ftp.excite.com/pub/jack/Excite_Log_12201999.gz*

TREC Implementation

We implement an ad-hoc search algorithm based on some of the top performing algorithms in use at TREC. The details of our implementation are discussed in Section 4. To be fair and to make sure that our implementation of the TREC ad-hoc algorithm is not broken, we test our implementation on several TREC ad-hoc tasks and verify that it is indeed state-of-the-art (see Section 4). Our evaluation procedure needs to accommodate the fact that the TREC ad-hoc algorithms retrieve pages, not sites, whereas most of the queries used in this study seek particular sites. Our implementation, like any other keyword-based system, has a tendency to retrieve multiple pages from a site. To have a site-oriented retrieval, we group the pages by the host they reside on and select the top twenty sites for evaluation. A site/host is given the same rank as the rank of the best page residing on that site. This is in-line with what many commercial search engines do.

Evaluation Measures

We compare the effectiveness of our implementation of TREC ad-hoc algorithm to four commercial search engines: Excite, Google, Lycos, and AltaVista Raging. For a given query, if a page is not found in the top ten ranks by a search engine, that engine gets no credit for that particular query. The assumption here is that if a user can't find a page in the top result page, the user will simply give up. This assumption is strongly supported by the fact that almost 85% of users don't request beyond just the first results screens for their query [18]. For every system we count the number of queries for which it retrieves the desired site at rank-1, up to rank-2, up to rank-3, and so on, and plot this on a graph (see Section 5). The higher the number of queries for which an engine retrieves the desired site at a certain rank, the better is the engine.

Using the top ten pages per query also allowed us to manually judge every run. Even though it is simple in principle to find if two URLs will get you the same page, in light of redirections (via the *refresh* HTML meta-tag), pages generated by javascripts, mirror sites, etc., this becomes a non-trivial exercise in the current web environment. Therefore, we check all the results by hand to find ranks of the relevant pages retrieved (as they may be retrieved under a completely different URL).

Collection of Pages

To eliminate the problems associated with the collection of web pages used in previous studies (see problems 3 and 5 in Section 2), we run our TREC ad-hoc algorithm over 217.5 gigabytes of freshly crawled web data (crawled between October 14–17, 2000) containing 17.8 million web pages. The assumption is that the commercial web search engines also have the same (fresh) copy of the pages crawled. The objective is to make sure that the underlying collection available to the our TREC algorithm is similar to the collection used by the commercial engines.

Since just 217.5 gigabytes of web data will not contain all the pages indexed by the commercial search engines, the TREC algorithm might be at a disadvantage because of the poor coverage of our crawl. To eliminate this problem, for every query in our test set, we add to our crawl all missing pages that are retrieved in the top ten ranks by any of the commercial search engines. We ran these queries on the commercial engines on October 17, 2000 and gathered the first ten results for each. We then fetched the pages that were not in our crawl and added them to our collection. This inclusion ensures that the TREC algorithms have access to all pages that have been retrieved by a commercial engine and are not at any disadvantage due to our small crawl. Even though quite unlikely, it is possible that we might have crawled pages that are not indexed by the commercial engines. This gives a slight advantage to the TREC ad-hoc algorithm in its ability to find such pages.

4. TREC AD-HOC ALGORITHM

Different groups participating in TREC have developed several ad-hoc algorithms over years. Most groups have their own expertise built into these algorithms. An analysis of some of the best performing TREC algorithms shows that the top ad-hoc algorithms at TREC have the following two common features: [23, 24]

1. Most of the top performing systems use a modern term weighting method developed in either the Okapi system [12, 13] or the SMART system [19, 20].

2. Most groups use a two-pass pseudo-feedback based query-expansion approach. In this approach a first pass retrieval is done to find a set of top (say) 10 or 20 documents related to the query, the query is expanded by adding new words/phrases from these documents using relevance feedback [14, 15], and this expanded query is used to generate the final ranking in a second-pass retrieval.

Each participating group has its own twist on the above two components. For example, several groups use collection enrichment [10], in which a much larger document collection is used in the first pass (instead of the target collection) to locate documents for use in the query-expansion process. In yet another enhancement, several groups assume that poorly ranked documents from the first pass are not relevant to the query and use this evidence of non-relevance in the query expansion process [20].

We implement an algorithm which is a scaled-down version of the ad-hoc algorithm used by Singhal et al. in [20]. As described in [23], this algorithm was one of the best performing ad-hoc algorithms at TREC-7. Here are the steps implemented in our algorithm.

- **Pass-1**: Using *dtn* queries and *dnb* documents, a first-pass retrieval is done (see Table 1 for an explanation of this term-weighting jargon).

- *Expansion:* Top ten (distinct) documents retrieved in the first pass are *assumed* to be relevant to the query. Rocchio's method (with parameters $\alpha = 1.0$, $\beta = 0.5$, and the γ factor is not needed here since we do not assume any documents as non-relevant) is used to expand the query by adding twenty new words with highest Rocchio weights [14]. To include the *idf*-factor in the expansion process, documents are *dtb* weighted.

- **Pass-2:** The expanded query is used with *dnb* documents to generate the final ranking.

Since web collections do have a reasonable number of duplicate documents, to do the query expansion well for the

d tf factor:

$$1 + ln(1 + ln(tf)) \qquad\qquad 0 \; if \; tf = 0$$

t idf factor:

$$log(\frac{N+1}{df})$$

b pivoted byte length normalization factor:

$$\frac{1}{0.8 + 0.2 \times \frac{length\ of\ document\ (in\ bytes)}{average\ document\ length\ (in\ bytes)}}$$

tf is the term's frequency in text (query/document)
N is the total number of documents in the collection
df is the number of documents that contain the term, and
the average document length depends on the collection.

dnb weighting: **d** factor × **b** factor
dtb weighting: **d** factor × **t** factor × **b** factor
dtn weighting: **d** factor × **t** factor

Table 1: Term Weighting Schemes

web collections, we have observed that we need to eliminate duplicate documents from the top ten documents used for query expansion. To do this, we retrieve top 100 documents in the first pass, and starting from rank 2 we test if a retrieved document is a duplicate of a previously ranked document. If it is, we remove it from the list. We do this until we get ten distinct documents. Two documents are considered duplicates of each other if they share more than 70% of their vocabulary. We have found this to be a reasonable heuristic for web pages.

To test that our implementation of this TREC ad-hoc algorithm is not broken and is indeed state-of-the-art, we run our system on two recent TREC ad-hoc tasks and compare its precision to the best performing systems at TREC. Since most web queries are short, we want to evaluate the system performance for short queries, and only use the 2-3 words *title* portion of the TREC queries. Our objective in this study is to do a precision oriented evaluation, we only compare systems based the precision in top ranks. We compare the precision of our system at rank 10 and at rank 20 to corresponding values for the five best performing systems at TREC using title-only queries (these values are available from the detailed results presented in the TREC proceedings, see Appendix A in [21] and [22]). The results are shown in Tables 2 and 3.

Tables 2 and 3 show the precision value for the best TREC systems ordered by decreasing performance. Inserted in that order, is the corresponding precision value for our system. These results show that our system, motivated by a state-of-the-art TREC ad-hoc algorithm is quite competitive with the top performing TREC systems. This is especially true considering the performance gap between the best and the fifth-best system is not very significant. For example, Table 2 indicates that for the TREC-7 ad-hoc task, the best performing system `ok7as` retrieves on an average 4.86 relevant documents in top 10 for a query, whereas the fifth-best performing system retrieves 4.28. That difference is not very large from a user's perspective.

In summary, these results verify that our implementation of a modern TREC ad-hoc algorithm is not broken and is indeed state-of-the-art. It will be reasonable to say that this system, when run over our fresh web collection, would produce results that will be quite comparable to the results produced by any other top TREC ad-hoc system.

5. RESULTS AND DISCUSSION

As described in Section 3, we use 100 queries in this study that seek a certain web page/site. These queries vary from finding company web sites, *e.g.*, "jordanian airlines", "Volkswagon", to finding college home pages as in "Walla Walla College", "Brigham Young University", to finding individual pages, *e.g.*, "mari ostendorf", "Vangelis Natsios", and so on. The results from both the first pass (no query expansion) and the second pass (with query expansion) of our TREC algorithm are compared to four commercial search engines: Excite, Google, Lycos, and AltaVista Raging.

Figure 1 shows the results of our experiments. The x-axis of Figure 1 shows the rank at which the desired site was retrieved by a system. The y-axis shows the cumulative number of queries for which the desired site was retrieved at or before the corresponding rank on the x-axis. For example, a point <6,82> on the plot indicates that the corresponding search engine retrieved the desired site at rank 6 or better for 82 out of the 100 queries. The higher the plot, the better the engine is. For example, the best engine (Engine 4) retrieves the relevant page at rank 1 for 81 out of the 100 queries. Whereas our two-pass TREC algorithm retrieves the relevant page at rank 1 for only 22 out of the 100 queries.

We would like to emphasize that for the TREC algorithms, any page that resides on the same host as the relevant page is counted as relevant. We assume that it should be fairly simple to do the host based grouping and present the root page for the host. This assumption might not always be true and the results presented in Figure 1 for the

System	**P@10**	System	**P@20**
ok7as	48.6%	ok7as	42.5%
OUR Implementation	**46.8%**	LNaTit7	39.1%
LNaTit7	46.2%	**OUR Implementation**	**38.8%**
pirc8At	44.8%	pirc8At	37.7%
att98atc	44.2%	FLab7at	37.5%
FLab7at	42.8%	att98atc	36.3%

Table 2: Precision at 10 and 20, TREC-7 ad-hoc task

System	**P@10**	System	**P@20**
ok8asxc	48.8%	pir9At0	44.1%
FLab8at	48.6%	FLab8at	42.6%
uwmt8a1	48.2%	uwmt8a1	42.5%
OUR Implementation	**48.0%**	**OUR Implementation**	**42.4%**
pir9At0	48.0%	att99ate	42.0%
att99ate	47.6%	ok8asxc	41.6%

Table 3: Precision at 10 and 20, TREC-8 ad-hoc task

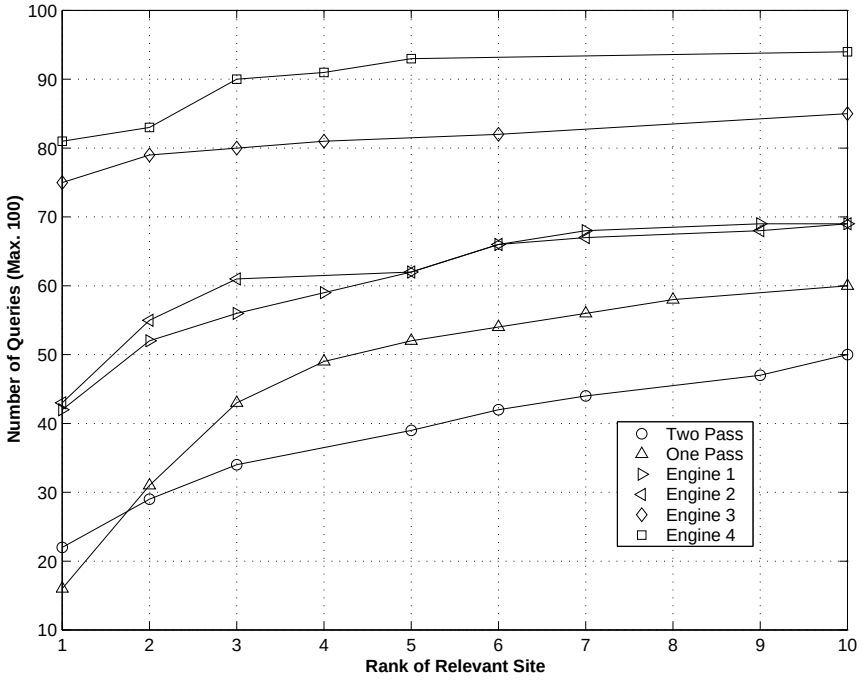

Figure 1: Performance of TREC algorithm compared to four commercial search engines.

TREC algorithms are, in some sense, best-case. Despite the best-case scenario for the TREC algorithms, Figure 1 shows that the TREC algorithms are far behind these four commercial web search engines for the kind of queries used in this study. The best commercial engine finds the relevant page in top 10 for 94/100 queries whereas the better performing (one-pass) TREC algorithm finds the relevant page for only 60 of the 100 queries. In short, the performance analysis presented in Figure 1 shows that the results from the TREC algorithms are consistently and notably below even the poorest of the commercial web search engines used in our study. This indicates that best TREC ad-hoc algorithms are by no means state-of-the-art for web search if our objective is to find a specific web site. These results contradict the results presented in [5, 6, 7]. However, we should say that these previous studies do not use the type of queries used in our study.

An even more surprising result is that adding a more complex query-expansion second pass does not improve the results, instead it makes the results somewhat worse. Using query expansion and doing two-pass retrieval we only find the relevant page for 50/100 queries in top 10 results as opposed to 60 pages for the one-pass algorithm. This result is in direct contradiction to the results obtained by TREC participants for the TREC ad-hoc benchmark tasks. In those results, it has been widely shown that in terms of average precision, which is how results are measured at TREC, a two-pass algorithm is almost always notably better than using just the first pass.

Use of RealNames

As an un-anticipated consequence of our choice of the type of queries used in this study, any engine that uses the RealNames navigation service [2] will have an edge in this test. RealNames links queries of this type to the true home-pages for the corresponding organization. On inspection of the results, we found that the best two engines in Figure 1 (Engine 4 and Engine 3) both use the RealNames service. The other two might be using it but there is no way for us to verify that by just looking at the results page. The best engine in Figure 1, Engine 4, used RealNames for 28 out of the 100 queries whereas the second-best engine, Engine 3, used RealNames for 36 queries.

This definitely gives an edge to these two engines. Unfortunately, we can't find out how the relevant pages would have been ranked by these engines if they were not using the RealNames service. Also, since our main objective is to compare these engines to the TREC algorithms, even if we remove the queries for which RealNames was used, the commercial engines still have a very large lead over the TREC algorithms, and the results from our experiments will stand. In all, it is quite safe to say that the commercial engines are using algorithms that are more effective for the type of queries used in this study.

Discussion

Analyzing some of the queries for which the TREC algorithms fail, we find that the most common reason for their failure is the presence of the query words with high frequency in non-relevant pages. For example, for the query "laguardia airport", the top ranked page (for the one-pass algorithm) is the flight schedule page for Tompkins County Airport (in Ithaca, NY, USA). This flight schedule contains the query word "laguardia" some ten times and gets a very high tf×idf based score. Similarly, for the query "american Kennel club", the top ranked page is a list of dog clubs, many of which have the query words in them. This list resides on the site `doghobbyist.com`. This is an obvious problem with keyword-based ranking systems, and we do see this problem hurting the results from our TREC algorithms.

On an in-depth examination, we notice why the more expensive two-pass system is worse than the one-pass system. For example, consider the query "horizon blue cross blue shield"[5]. The one-pass system retrieves the relevant page, `www.bcbsnj.com`, as the top ranked page. However, the first pass also retrieves many health insurance/care related pages in the top ten pages. In the query expansion step, this query loses its focus on "horizon blue cross blue shield" and instead becomes a general health insurance query, failing to retrieve the desired page in the second pass. This loss of focus is observed for many other queries in our set.

It is worth noting that many pages retrieved by the TREC algorithms are quite relevant to the topic at hand. They are just not the page the user was looking for in our experiments. Under the TREC criteria for judging relevance, many of these pages are "on topic" and will be judged relevant. This would explain why under the TREC measurements, the commercial engines do not show any advantage over the TREC algorithms.

6. CONCLUSIONS

Searching the web accurately is becoming increasingly critical as the web grows. In this study we have revisited the question if link-based methods hold any advantage over state-of-the-art keyword-based methods for searching a web collection. For the type of queries used in the study: finding the web site of an entity, we observe that commercial search engines that use some link-based ranking schemes outperform a modern keyword-based algorithm by a large margin. Such queries are quite prevalent in web search. The results from this study establish, for the first time, that for a certain type of queries, link-based ranking algorithms are indeed better than using a modern keyword-based algorithm. Most previous studies that have done this comparison tend to show otherwise. It would be interesting to extend this work to other types of queries as well, for example to the queries that seek high quality web sites on a certain topic.

7. REFERENCES

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, pages 107–117, April 1998.

[2] RealNames Corporation. Realnames and iwon partner to simplify internet navigation. Press Release, `http://www.realnames.com`.

[3] W. Goffman. On relevance as a measure. *Information Storage and Retrieval*, 2(3):201–203, 1964.

[4] M. Gordon and P. Pathak. Finding information on the world wide web: the retrieval effectiveness of search engines. *Information Processing and Management*, 25(2):141–180, 1999.

[5] This is the New Jersey arm of Blue Cross Blue Shield, a big American health insurance company.

[5] D. Hawking, N. Craswell, and P. Thistlewaite. Overview of the TREC-7 very large collection track. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 91–104. NIST Special Publication 500-242, July 1999.

[6] D. Hawking, N. Craswell, P. Thistlewaite, and D. Harman. Results and challenges in web search evaluation. In *Proceedings of the Eighth International World Wide Web Conference*, pages 243–252, May 1999.

[7] D. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 web track. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 131–150. NIST Special Publication 500-246, 2000.

[8] B. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: a study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.

[9] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACL-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[10] K.L. Kwok. Improving two-stage ad-hoc retrieval for short queries. In *Proceedings of the Twenty-First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 250–256. Association for Computing Machinery, New York, August 1998.

[11] H. Vernon Leighton and J. Srivastava. First 20 precision among world web search services (search engines). *Journal of the American Society for Information Science*, 50(10):870–881, 1999.

[12] S. Roberstson, S. Walker, and M. Beaulieu. Okapi at TREC-7: automatic ad-hoc, filtering, VLC and interactive track. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 253–264. NIST Special Publication 500-242, July 1999.

[13] S. Robertson and S. Walker. Some simple effective approximations to the 2–poisson model for probabilistic weighted retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241. Springer-Verlag, New York, July 1994.

[14] J.J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System—Experiments in Automatic Document Processing*, pages 313–323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.

[15] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[16] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill Book Co., New York, 1983.

[17] J. Savoy and J. Picard. Report on the TREC-8 experiment: Searching on the web and in distributed collections. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 229–240. NIST Special Publication 500-246, 2000.

[18] C. Silverstein, M. Henzinger, J. Marais, and M. Moricz. Analysis of a very large AltaVista query log. Technical Report TR 1998-014, Compaq Systems Research Center, Palo Alto, CA, 1998.

[19] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29. Association for Computing Machinery, New York, August 1996.

[20] A. Singhal, J. Choi, D. Hindle, D. Lewis, and F. Pereira. AT&T at TREC-7. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 239–252. NIST Special Publication 500-242, July 1999.

[21] E. M. Voorhees and D. K. Harman, editors. *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication 500-242, July 1999.

[22] E. M. Voorhees and D. K. Harman, editors. *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*. NIST Special Publication 500-246, 2000.

[23] E.M. Voorhees and D.K. Harman. Overview of the seventh Text REtrieval Conference (TREC-7). In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 1–24. NIST Special Publication 500-242, July 1999.

[24] E.M. Voorhees and D.K. Harman. Overview of the eighth Text REtrieval Conference (TREC-8). In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 1–24, 2000.

[25] Search Engine Watch. `http://www.searchenginewatch.com.`