HITS on the Web: How does it Compare?

Marc Najork
Microsoft Research
1065 La Avenida
Mountain View, CA, USA
najork@microsoft.com

Hugo Zaragoza[*]
Yahoo! Research Barcelona
Ocata 1
Barcelona 08003, Spain
hugoz@es.yahoo-inc.com

Michael Taylor
Microsoft Research
7 J J Thompson Ave
Cambridge CB3 0FB, UK
mitaylor@microsoft.com

ABSTRACT

This paper describes a large-scale evaluation of the effectiveness of HITS in comparison with other link-based ranking algorithms, when used in combination with a state-of-the-art text retrieval algorithm exploiting anchor text. We quantified their effectiveness using three common performance measures: the mean reciprocal rank, the mean average precision, and the normalized discounted cumulative gain measurements. The evaluation is based on two large data sets: a breadth-first search crawl of 463 million web pages containing 17.6 billion hyperlinks and referencing 2.9 billion distinct URLs; and a set of 28,043 queries sampled from a query log, each query having on average 2,383 results, about 17 of which were labeled by judges. We found that HITS outperforms PageRank, but is about as effective as web-page in-degree. The same holds true when any of the link-based features are combined with the text retrieval algorithm. Finally, we studied the relationship between query specificity and the effectiveness of selected features, and found that link-based features perform better for general queries, whereas BM25F performs better for specific queries.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Storage and Retrieval—*search process, selection process*

General Terms

Algorithms, Measurement, Experimentation

Keywords

Ranking, PageRank, HITS, BM25F, MRR, MAP, NDCG

1. INTRODUCTION

Link graph features such as in-degree and PageRank have been shown to significantly improve the performance of text retrieval algorithms on the web. The HITS algorithm is also believed to be of interest for web search; to some degree, one may expect HITS to be more informative that other link-based features because it is query-dependent: it tries to measure the interest of pages with respect to a given query. However, it remains unclear today whether there are practical benefits of HITS over other link graph measures. This is even more true when we consider that modern retrieval algorithms used on the web use a document representation which incorporates the document's anchor text, *i.e.* the text of incoming links. This, at least to some degree, takes the link graph into account, in a query-dependent manner.

Comparing HITS to PageRank or in-degree empirically is no easy task. There are two main difficulties: scale and relevance. Scale is important because link-based features are known to improve in quality as the document graph grows. If we carry out a small experiment, our conclusions won't carry over to large graphs such as the web. However, computing HITS efficiently on a graph the size of a realistic web crawl is extraordinarily difficult. Relevance is also crucial because we cannot measure the performance of a feature in the absence of human judgments: what is crucial is ranking at the top of the ten or so documents that a user will peruse. To our knowledge, this paper is the first attempt to evaluate HITS at a large scale and compare it to other link-based features with respect to human evaluated judgment.

Our results confirm many of the intuitions we have about link-based features and their relationship to text retrieval methods exploiting anchor text. This is reassuring: in the absence of a theoretical model capable of tying these measures with relevance, the only way to validate our intuitions is to carry out realistic experiments. However, we were quite surprised to find that HITS, a query-dependent feature, is about as effective as web page in-degree, the most simple-minded query-independent link-based feature. This continues to be true when the link-based features are combined with a text retrieval algorithm exploiting anchor text.

The remainder of this paper is structured as follows: Section 2 surveys related work. Section 3 describes the data sets we used in our study. Section 4 reviews the performance measures we used. Sections 5 and 6 describe the PageRank and HITS algorithms in more detail, and sketch the computational infrastructure we employed to carry out large scale experiments. Section 7 presents the results of our evaluations, and Section 8 offers concluding remarks.

[*] This work was performed while the author worked for Microsoft Research.

SIGIR '07, July 23–27, 2007, Amsterdam, The Netherlands.
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...$5.00.

2. RELATED WORK

The idea of using hyperlink analysis for ranking web search results arose around 1997, and manifested itself in the HITS [16, 17] and PageRank [5, 21] algorithms. The popularity of these two algorithms and the phenomenal success of the Google search engine, which uses PageRank, have spawned a large amount of subsequent research.

There are numerous attempts at improving the effectiveness of HITS and PageRank. Query-dependent link-based ranking algorithms inspired by HITS include SALSA [19], Randomized HITS [20], and PHITS [7], to name a few. Query-independent link-based ranking algorithms inspired by PageRank include TrafficRank [22], BlockRank [14], and TrustRank [11], and many others.

Another line of research is concerned with analyzing the mathematical properties of HITS and PageRank. For example, Borodin *et al.* [3] investigated various theoretical properties of PageRank, HITS, SALSA, and PHITS, including their similarity and stability, while Bianchini *et al.* [2] studied the relationship between the structure of the web graph and the distribution of PageRank scores, and Langville and Meyer examined basic properties of PageRank such as existence and uniqueness of an eigenvector and convergence of power iteration [18].

Given the attention that has been paid to improving the effectiveness of PageRank and HITS, and the thorough studies of the mathematical properties of these algorithms, it is somewhat surprising that very few evaluations of their effectiveness have been published. We are aware of two studies that have attempted to formally evaluate the effectiveness of HITS and of PageRank. Amento *et al.* [1] employed quantitative measures, but based their experiments on the result sets of just 5 queries and the web-graph induced by topical crawls around the result set of each query. A more recent study by Borodin *et al.* [4] is based on 34 queries, result sets of 200 pages per query obtained from Google, and a neighborhood graph derived by retrieving 50 in-links per result from Google. By contrast, our study is based on over 28,000 queries and a web graph covering 2.9 billion URLs.

3. OUR DATA SETS

Our evaluation is based on two data sets: a large web graph and a substantial set of queries with associated results, some of which were labeled by human judges.

Our web graph is based on a web crawl that was conducted in a breadth-first-search fashion, and successfully retrieved 463,685,607 HTML pages. These pages contain 17,672,011,890 hyperlinks (after eliminating duplicate hyperlinks embedded in the same web page), which refer to a total of 2,897,671,002 URLs. Thus, at the end of the crawl there were 2,433,985,395 URLs in the “frontier” set of the crawler that had been discovered, but not yet downloaded. The mean out-degree of crawled web pages is 38.11; the mean in-degree of discovered pages (whether crawled or not) is 6.10. Also, it is worth pointing out that there is a lot more variance in in-degrees than in out-degrees; some popular pages have millions of incoming links. As we will see, this property affects the computational cost of HITS.

Our query set was produced by sampling 28,043 queries from the MSN Search query log, and retrieving a total of 66,846,214 result URLs for these queries (using commercial search engine technology), or about 2,838 results per query on average. It is important to point out that our 2.9 billion URL web graph does not cover all these result URLs. In fact, only 9,525,566 of the result URLs (about 14.25%) were covered by the graph.

485,656 of the results in the query set (about 0.73% of all results, or about 17.3 results per query) were rated by human judges as to their relevance to the given query, and labeled on a six-point scale (the labels being “definitive”, “excellent”, “good”, “fair”, “bad” and “detrimental”). Results were selected for judgment based on their commercial search engine placement; in other words, the subset of labeled results is not random, but biased towards documents considered relevant by pre-existing ranking algorithms.

Involving a human in the evaluation process is extremely cumbersome and expensive; however, human judgments are crucial for the evaluation of search engines. This is so because no document features have been found yet that can effectively estimate the relevance of a document to a user query. Since content-match features are very unreliable (and even more so link features, as we will see) we need to ask a human to evaluate the results in order to compare the quality of features.

Evaluating the retrieval results from document scores and human judgments is not trivial and has been the subject of many investigations in the IR community. A good performance measure should correlate with user satisfaction, taking into account that users will dislike having to delve deep in the results to find relevant documents. For this reason, standard correlation measures (such as the correlation coefficient between the score and the judgment of a document), or order correlation measures (such as Kendall tau between the score and judgment induced orders) are not adequate.

4. MEASURING PERFORMANCE

In this study, we quantify the effectiveness of various ranking algorithms using three measures: NDCG, MRR, and MAP.

The *normalized discounted cumulative gains* (NDCG) measure [13] discounts the contribution of a document to the overall score as the document's rank increases (assuming that the best document has the lowest rank). Such a measure is particularly appropriate for search engines, as studies have shown that search engine users rarely consider anything beyond the first few results [12]. NDCG values are normalized to be between 0 and 1, with 1 being the NDCG of a “perfect” ranking scheme that completely agrees with the assessment of the human judges. The discounted cumulative gain at a particular rank-threshold T ($DCG@T$) is defined to be $\sum_{j=1}^{T} \frac{1}{\log(1+j)} \left(2^{r(j)} - 1\right)$, where $r(j)$ is the rating (0=detrimental, 1=bad, 2=fair, 3=good, 4=excellent, and 5=definitive) at rank j. The NDCG is computed by dividing the DCG of a ranking by the highest possible DCG that can be obtained for that query. Finally, the NDGCs of all queries in the query set are averaged to produce a mean NDCG.

The *reciprocal rank* (RR) of the ranked result set of a query is defined to be the reciprocal value of the rank of the highest-ranking relevant document in the result set. The RR at rank-threshold T is defined to be 0 if none of the highest-ranking T documents is relevant. The *mean reciprocal rank* (MRR) of a query set is the average reciprocal rank of all queries in the query set.

Given a ranked set of n results, let $rel(i)$ be 1 if the result at rank i is relevant and 0 otherwise. The *precision* $P(j)$ at rank j is defined to be $\frac{1}{j}\sum_{i=1}^{j} rel(i)$, *i.e.* the fraction of the relevant results among the j highest-ranking results. The *average precision* (AP) at rank-threshold k is defined to be $\frac{\sum_{i=1}^{k} P(i) rel(i)}{\sum_{i=1}^{n} rel(i)}$. The *mean average precision* (MAP) of a query set is the mean of the average precisions of all queries in the query set.

The above definitions of MRR and MAP rely on the notion of a "relevant" result. We investigated two definitions of relevance: One where all documents rated "fair" or better were deemed relevant, and one were all documents rated "good" or better were deemed relevant. For reasons of space, we only report MAP and MRR values computed using the latter definition; using the former definition does not change the qualitative nature of our findings. Similarly, we computed NDCG, MAP, and MRR values for a wide range of rank-thresholds; we report results here at rank 10; again, changing the rank-threshold never led us to different conclusions.

Recall that over 99% of documents are unlabeled. We chose to treat all these documents as irrelevant to the query. For some queries, however, not all relevant documents have been judged. This introduces a bias into our evaluation: features that bring new documents to the top of the rank may be penalized. This will be more acute for features less correlated to the pre-existing commercial ranking algorithms used to select documents for judgment. On the other hand, most queries have few perfect relevant documents (*i.e.* home page or item searches) and they will most often be within the judged set.

5. COMPUTING PAGERANK ON A LARGE WEB GRAPH

PageRank is a query-independent measure of the *importance* of web pages, based on the notion of peer-endorsement: A hyperlink from page A to page B is interpreted as an endorsement of page B's content by page A's author. The following recursive definition captures this notion of endorsement:

$$R(v) = \sum_{(u,v)\in E} \frac{R(u)}{Out(u)}$$

where $R(v)$ is the score (importance) of page v, (u, v) is an edge (hyperlink) from page u to page v contained in the edge set E of the web graph, and $Out(u)$ is the out-degree (number of embedded hyperlinks) of page u. However, this definition suffers from a severe shortcoming: In the fixed-point of this recursive equation, only edges that are part of a strongly-connected component receive a non-zero score. In order to overcome this deficiency, Page *et al.* grant each page a guaranteed "minimum score", giving rise to the definition of standard PageRank:

$$R(v) = \frac{d}{|V|} + (1-d) \sum_{(u,v)\in E} \frac{R(u)}{Out(u)}$$

where $|V|$ is the size of the vertex set (the number of known web pages), and d is a "damping factor", typically set to be between 0.1 and 0.2.

Assuming that scores are normalized to sum up to 1, PageRank can be viewed as the stationary probability distribution of a random walk on the web graph, where at each step of the walk, the walker with probability $1 - d$ moves from its current node u to a neighboring node v, and with probability d selects a node uniformly at random from all nodes in the graph and jumps to it. In the limit, the random walker is at node v with probability $R(v)$.

One issue that has to be addressed when implementing PageRank is how to deal with "sink" nodes, nodes that do not have any outgoing links. One possibility would be to select another node uniformly at random and transition to it; this is equivalent to adding edges from each sink nodes to all other nodes in the graph. We chose the alternative approach of introducing a single "phantom" node. Each sink node has an edge to the phantom node, and the phantom node has an edge to itself.

In practice, PageRank scores can be computed using power iteration. Since PageRank is query-independent, the computation can be performed off-line ahead of query time. This property has been key to PageRank's success, since it is a challenging engineering problem to build a system that can perform any non-trivial computation on the web graph at query time.

In order to compute PageRank scores for all 2.9 billion nodes in our web graph, we implemented a distributed version of PageRank. The computation consists of two distinct phases. In the first phase, the link files produced by the web crawler, which contain page URLs and their associated link URLs in textual form, are partitioned among the machines in the cluster used to compute PageRank scores, and converted into a more compact format along the way. Specifically, URLs are partitioned across the machines in the cluster based on a hash of the URLs' host component, and each machine in the cluster maintains a table mapping the URL to a 32-bit integer. The integers are drawn from a densely packed space, so as to make suitable indices into the array that will later hold the PageRank scores. The system then translates our log of pages and their associated hyperlinks into a compact representation where both page URLs and link URLs are represented by their associated 32-bit integers. Hashing the host component of the URLs guarantees that all URLs from the same host are assigned to the same machine in our scoring cluster. Since over 80% of all hyperlinks on the web are relative (that is, are between two pages on the same host), this property greatly reduces the amount of network communication required by the second stage of the distributed scoring computation.

The second phase performs the actual PageRank power iteration. Both the link data and the current PageRank vector reside on disk and are read in a streaming fashion; while the new PageRank vector is maintained in memory. We represent PageRank scores as 64-bit floating point numbers. PageRank contributions to pages assigned to remote machines are streamed to the remote machine via a TCP connection.

We used a three-machine cluster, each machine equipped with 16 GB of RAM, to compute standard PageRank scores for all 2.9 billion URLs that were contained in our web graph. We used a damping factor of 0.15, and performed 200 power iterations. Starting at iteration 165, the L_∞ norm of the change in the PageRank vector from one iteration to the next had stopped decreasing, indicating that we had reached as much of a fixed point as the limitations of 64-bit floating point arithmetic would allow.

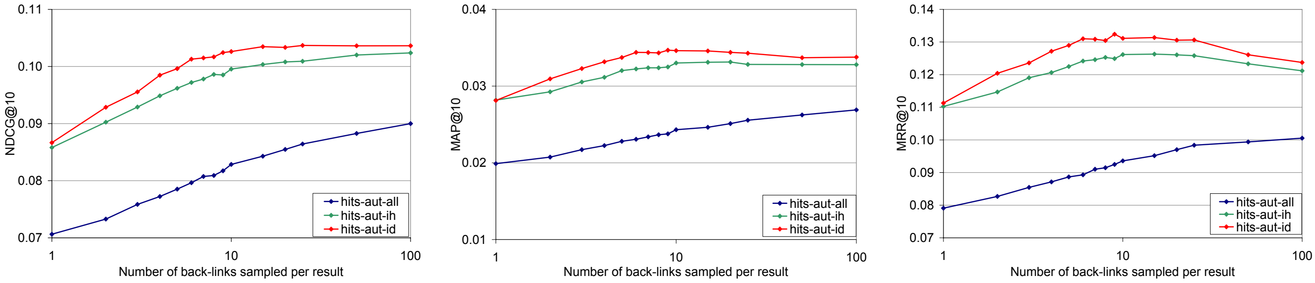

Figure 1: Effectiveness of authority scores computed using different parameterizations of HITS.

A post-processing phase uses the final PageRank vectors (one per machine) and the table mapping URLs to 32-bit integers (representing indices into each PageRank vector) to score the result URL in our query log. As mentioned above, our web graph covered 9,525,566 of the 66,846,214 result URLs. These URLs were annotated with their computed PageRank score; all other URLs received a score of 0.

6. HITS

HITS, unlike PageRank, is a query-dependent ranking algorithm. HITS (which stands for "Hypertext Induced Topic Search") is based on the following two intuitions: First, hyperlinks can be viewed as topical endorsements: A hyperlink from a page u devoted to topic T to another page v is likely to endorse the authority of v *with respect to topic* T. Second, the result set of a particular query is likely to have a certain amount of topical coherence. Therefore, it makes sense to perform link analysis not on the entire web graph, but rather on just the neighborhood of pages contained in the result set, since this neighborhood is more likely to contain topically relevant links. But while the set of nodes immediately reachable from the result set is manageable (given that most pages have only a limited number of hyperlinks embedded into them), the set of pages immediately *leading to* the result set can be enormous. For this reason, Kleinberg suggests sampling a fixed-size random subset of the pages linking to any high-indegree page in the result set. Moreover, Kleinberg suggests considering only links that cross host boundaries, the rationale being that links between pages on the same host ("intrinsic links") are likely to be navigational or nepotistic and not topically relevant.

Given a web graph (V, E) with vertex set V and edge set $E \subseteq V \times V$, and the set of result URLs to a query (called the *root set* $R \subseteq V$) as input, HITS computes a neighborhood graph consisting of a *base set* $B \subseteq V$ (the root set and some of its neighboring vertices) and some of the edges in E induced by B. In order to formalize the definition of the neighborhood graph, it is helpful to first introduce a sampling operator and the concept of a link-selection predicate.

Given a set A, the notation $\mathcal{S}_n[A]$ draws n elements uniformly at random from A; $\mathcal{S}_n[A] = A$ if $|A| \leq n$.

A *link section predicate* P takes an edge $(u, v) \in E$. In this study, we use the following three link section predicates:

$$\begin{aligned} all(u,v) &\Leftrightarrow true \\ ih(u,v) &\Leftrightarrow host(u) \neq host(v) \\ id(u,v) &\Leftrightarrow domain(u) \neq domain(v) \end{aligned}$$

where $host(u)$ denotes the host of URL u, and $domain(u)$ denotes the domain of URL u. So, *all* is true for all links, whereas *ih* is true only for inter-host links, and *id* is true only for inter-domain links.

The *outlinked-set* O^P of the root set R w.r.t. a link-selection predicate P is defined to be:

$$O^P = \bigcup_{u \in R} \{v \in V : (u,v) \in E \wedge P(u,v)\}$$

The *inlinking-set* I_s^P of the root set R w.r.t. a link-selection predicate P and a sampling value s is defined to be:

$$I_s^P = \bigcup_{v \in R} \mathcal{S}_s[\{u \in V : (u,v) \in E \wedge P(u,v)\}]$$

The *base set* B_s^P of the root set R w.r.t. P and s is defined to be:

$$B_s^P = R \cup I_s^P \cup O^P$$

The *neighborhood graph* (B_s^P, N_s^P) has the base set B_s^P as its vertex set and an edge set N_s^P containing those edges in E that are covered by B_s^P and permitted by P:

$$N_s^P = \{(u,v) \in E : u \in B_s^P \wedge v \in B_s^P \wedge P(u,v)\}$$

To simplify notation, we write B to denote B_s^P, and N to denote N_s^P.

For each node u in the neighborhood graph, HITS computes two scores: an *authority score* $A(u)$, estimating how authoritative u is on the topic induced by the query, and a *hub score* $H(u)$, indicating whether u is a good reference to many authoritative pages. This is done using the following algorithm:

1. For all $u \in B$ do $H(u) := \sqrt{\frac{1}{|B|}}, A(u) := \sqrt{\frac{1}{|B|}}$.
2. Repeat until H and A converge:
   (a) For all $v \in B : A'(v) := \sum_{(u,v) \in N} H(u)$
   (b) For all $u \in B : H'(u) := \sum_{(u,v) \in N} A(v)$
   (c) $H := \|H'\|_2, A := \|A'\|_2$

where $\|X\|_2$ normalizes the vector X to unit length in euclidean space, *i.e.* the squares of its elements sum up to 1.

In practice, implementing a system that can compute HITS within the time constraints of a major search engine (where the peak query load is in the thousands of queries per second, and the desired query response time is well below one second) is a major engineering challenge. Among other things, the web graph cannot reasonably be stored on disk, since

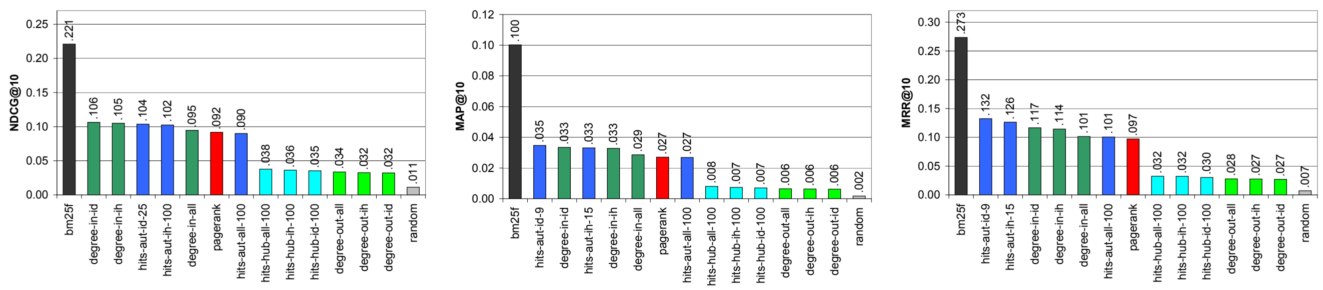

Figure 2: Effectiveness of different features.

seek times of modern hard disks are too slow to retrieve the links within the time constraints, and the graph does not fit into the main memory of a single machine, even when using the most aggressive compression techniques.

In order to experiment with HITS and other query-dependent link-based ranking algorithms that require non-regular accesses to arbitrary nodes and edges in the web graph, we implemented a system called the *Scalable Hyperlink Store*, or SHS for short. SHS is a special-purpose database, distributed over an arbitrary number of machines that keeps a highly compressed version of the web graph in memory and allows very fast lookup of nodes and edges. On our hardware, it takes an average of 2 microseconds to map a URL to a 64-bit integer handle called a UID, 15 microseconds to look up all incoming or outgoing link UIDs associated with a page UID, and 5 microseconds to map a UID back to a URL (the last functionality not being required by HITS). The RPC overhead is about 100 microseconds, but the SHS API allows many lookups to be batched into a single RPC request.

We implemented the HITS algorithm using the SHS infrastructure. We compiled three SHS databases, one containing all 17.6 billion links in our web graph (*all*), one containing only links between pages that are on different hosts (*ih*, for "inter-host"), and one containing only links between pages that are on different domains (*id*). We consider two URLs to belong to different hosts if the host portions of the URLs differ (in other words, we make no attempt to determine whether two distinct symbolic host names refer to the same computer), and we consider a domain to be the name purchased from a registrar (for example, we consider news.bbc.co.uk and www.bbc.co.uk to be different hosts belonging to the same domain). Using each of these databases, we computed HITS authority and hub scores for various parameterizations of the sampling operator $\mathcal{S}$, sampling between 1 and 100 back-links of each page in the root set. Result URLs that were not covered by our web graph automatically received authority and hub scores of 0, since they were not connected to any other nodes in the neighborhood graph and therefore did not receive any endorsements.

We performed forty-five different HITS computations, each combining one of the three link selection predicates (*all*, *ih*, and *id*) with a sampling value. For each combination, we loaded one of the three databases into an SHS system running on six machines (each equipped with 16 GB of RAM), and computed HITS authority and hub scores, one query at a time. The longest-running combination (using the *all* database and sampling 100 back-links of each root set vertex) required 30,456 seconds to process the entire query set, or about 1.1 seconds per query on average.

7. EXPERIMENTAL RESULTS

For a given query Q, we need to rank the set of documents satisfying Q (the "result set" of Q). Our hypothesis is that good features should be able to rank relevant documents in this set higher than non-relevant ones, and this should result in an increase in each performance measure over the query set. We are specifically interested in evaluating the usefulness of HITS and other link-based features. In principle, we could do this by sorting the documents in each result set by their feature value, and compare the resulting NDCGs. We call this ranking with *isolated features*.

Let us first examine the relative performance of the different parameterizations of the HITS algorithm we examined. Recall that we computed HITS for each combination of three link section schemes – all links (*all*), inter-host links only (*ih*), and inter-domain links only (*id*) – with back-link sampling values ranging from 1 to 100. Figure 1 shows the impact of the number of sampled back-links on the retrieval performance of HITS authority scores. Each graph is associated with one performance measure. The horizontal axis of each graph represents the number of sampled back-links, the vertical axis represents performance under the appropriate measure, and each curve depicts a link selection scheme. The *id* scheme slightly outperforms *ih*, and both vastly outperform the *all* scheme – eliminating nepotistic links pays off. The performance of the *all* scheme increases as more back-links of each root set vertex are sampled, while the performance of the *id* and *ih* schemes peaks at between 10 and 25 samples and then plateaus or even declines, depending on the performance measure.

Having compared different parameterizations of HITS, we will now fix the number of sampled back-links at 100 and compare the three link selection schemes against other isolated features: PageRank, in-degree and out-degree counting links of all pages, of different hosts only and of different domains only (*all, ih* and *id* datasets respectively), and a text retrieval algorithm exploiting anchor text: BM25F[24]. BM25F is a state-of-the art ranking function solely based on textual content of the documents and their associated anchor texts. BM25F is a descendant of BM25 that combines the different textual fields of a document, namely title, body and anchor text. This model has been shown to be one of the best-performing web search scoring functions over the last few years [8, 24]. BM25F has a number of free parameters (2 per field, 6 in our case); we used the parameter values described in [24].

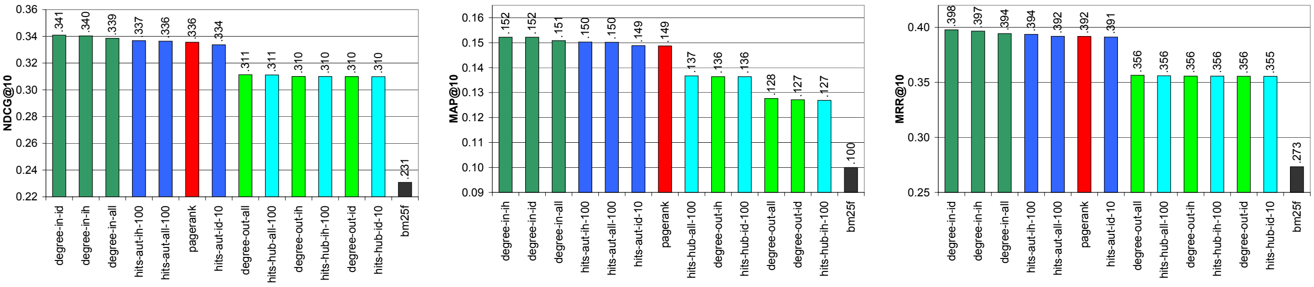

Figure 3: Effectiveness measures for linear combinations of link-based features with BM25F.

Figure 2 shows the NDCG, MRR, and MAP measures of these features. Again all performance measures (and for all rank-thresholds we explored) agree. As expected, BM25F outperforms all link-based features by a large margin. The link-based features are divided into two groups, with a noticeable performance drop between the groups. The better-performing group consists of the features that are based on the number and/or quality of incoming links (in-degree, PageRank, and HITS authority scores); and the worse-performing group consists of the features that are based on the number and/or quality of outgoing links (out-degree and HITS hub scores). In the group of features based on incoming links, features that ignore nepotistic links perform better than their counterparts using *all* links. Moreover, using only inter-domain (*id*) links seems to be marginally better than using inter-host (*ih*) links.

The fact that features based on outgoing links underperform those based on incoming links matches our expectations; if anything, it is mildly surprising that outgoing links provide a useful signal for ranking at all. On the other hand, the fact that in-degree features outperform PageRank under all measures is quite surprising. A possible explanation is that link-spammers have been targeting the published PageRank algorithm for many years, and that this has led to anomalies in the web graph that affect PageRank, but not other link-based features that explore only a distance-1 neighborhood of the result set. Likewise, it is surprising that simple query-independent features such as in-degree, which might estimate global quality but cannot capture relevance to a query, would outperform query-dependent features such as HITS authority scores.

However, we cannot investigate the effect of these features in isolation, without regard to the overall ranking function, for several reasons. First, features based on the textual content of documents (as opposed to link-based features) are the best predictors of relevance. Second, link-based features can be strongly correlated with textual features for several reasons, mainly the correlation between in-degree and number of textual anchor matches.

Feature	Transform function
bm25f	$T(s) = s$
pagerank	$T(s) = \log(s + 3 \cdot 10^{-12})$
degree-in-*	$T(s) = \log(s + 3 \cdot 10^{-2})$
degree-out-*	$T(s) = \log(s + 3 \cdot 10^{3})$
hits-aut-*	$T(s) = \log(s + 3 \cdot 10^{-8})$
hits-hub-*	$T(s) = \log(s + 3 \cdot 10^{-1})$

Table 1: Near-optimal feature transform functions.

Therefore, one must consider the effect of link-based features *in combination* with textual features. Otherwise, we may find a link-based feature that is very good in isolation but is strongly correlated with textual features and results in no overall improvement; and vice versa, we may find a link-based feature that is weak in isolation but significantly improves overall performance.

For this reason, we have studied the combination of the link-based features above with BM25F. All feature combinations were done by considering the linear combination of two features as a document score, using the formula $score(d) = \sum_{i=1}^{n} w_i T_i(F_i(d))$, where d is a document (or document-query pair, in the case of BM25F), $F_i(d)$ (for $1 \le i \le n$) is a feature extracted from d, T_i is a transform, and w_i is a free scalar weight that needs to be tuned. We chose transform functions that we empirically determined to be well-suited. Table 1 shows the chosen transform functions.

This type of linear combination is appropriate if we assume features to be independent with respect to relevance and an exponential model for link features, as discussed in [8]. We tuned the weights by selecting a random subset of 5,000 queries from the query set, used an iterative refinement process to find weights that maximized a given performance measure on that training set, and used the remaining 23,043 queries to measure the performance of the thus derived scoring functions.

We explored the pairwise combination of BM25F with every link-based scoring function. Figure 3 shows the NDCG, MRR, and MAP measures of these feature combinations, together with a baseline BM25F score (the right-most bar in each graph), which was computed using the same subset of 23,045 queries that were used as the test set for the feature combinations. Regardless of the performance measure applied, we can make the following general observations:

1. Combining any of the link-based features with BM25F results in a substantial performance improvement over BM25F in isolation.

2. The combination of BM25F with features based on incoming links (PageRank, in-degree, and HITS authority scores) performs substantially better than the combination with features based on outgoing links (HITS hub scores and out-degree).

3. The performance differences between the various combinations of BM25F with features based on incoming links is comparatively small, and the relative ordering of feature combinations is fairly stable across the dif-

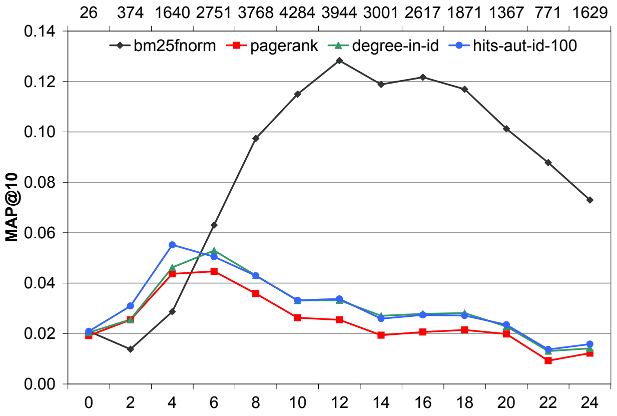

Figure 4: Effectiveness measures for selected isolated features, broken down by query specificity.

ferent performance measures used. However, the combination of BM25F with any in-degree variant, and in particular with *id* in-degree, consistently outperforms the combination of BM25F with PageRank or HITS authority scores, and can be computed much easier and faster.

Finally, we investigated whether certain features are better for some queries than for others. Particularly, we are interested in the relationship between the specificity of a query and the performance of different ranking features. The most straightforward measure of the specificity of a query Q would be the number of documents in a search engine's corpus that satisfy Q. Unfortunately, the query set available to us did not contain this information. Therefore, we chose to approximate the specificity of Q by summing up the inverse document frequencies of the individual query terms comprising Q. The inverse document frequency (IDF) of a term t with respect to a corpus C is defined to be $logN/doc(t)$, where $doc(t)$ is the number of documents in C containing t and N is the total number of documents in C. By summing up the IDFs of the query terms, we make the (flawed) assumption that the individual query terms are independent of each other. However, while not perfect, this approximation is at least directionally correct.

We broke down our query set into 13 buckets, each bucket associated with an interval of query IDF values, and we computed performance metrics for all ranking functions applied (in isolation) to the queries in each bucket. In order to keep the graphs readable, we will not show the performance of all the features, but rather restrict ourselves to the four most interesting ones: PageRank, *id* HITS authority scores, *id* in-degree, and BM25F. Figure 4 shows the MAP@10 for all 13 query specificity buckets. Buckets on the far left of each graph represent very general queries; buckets on the far right represent very specific queries. The figures on the upper x axis of each graph show the number of queries in each bucket (e.g. the right-most bucket contains 1,629 queries). BM25F performs best for medium-specific queries, peaking at the buckets representing the IDF sum interval [12,14). By comparison, HITS peaks at the bucket representing the IDF sum interval [4,6), and PageRank and in-degree peak at the bucket representing the interval [6,8), i.e. more general queries.

8. CONCLUSIONS AND FUTURE WORK

This paper describes a large-scale evaluation of the effectiveness of HITS in comparison with other link-based ranking algorithms, in particular PageRank and in-degree, when applied in isolation or in combination with a text retrieval algorithm exploiting anchor text (BM25F). Evaluation is carried out with respect to a large number of human evaluated queries, using three different measures of effectiveness: NDCG, MRR, and MAP. Evaluating link-based features in isolation, we found that web page in-degree outperforms PageRank, and is about as effwective as HITS authority scores. HITS hub scores and web page out-degree are much less effective ranking features, but still outperform a random ordering. A linear combination of any link-based features with BM25F produces a significant improvement in performance, and there is a clear difference between combining BM25F with a feature based on incoming links (in-degree, PageRank, or HITS authority scores) and a feature based on outgoing links (HITS hub scores and out-degree), but within those two groups the precise choice of link-based feature matters relatively little.

We believe that the measurements presented in this paper provide a solid evaluation of the best well-known link-based ranking schemes. There are many possible variants of these schemes, and many other link-based ranking algorithms have been proposed in the literature, hence we do not claim this work to be the last word on this subject, but rather the first step on a long road. Future work includes evaluation of different parameterizations of PageRank and HITS. In particular, we would like to study the impact of changes to the PageRank damping factor on effectiveness, the impact of various schemes meant to counteract the effects of link spam, and the effect of weighing hyperlinks differently depending on whether they are nepotistic or not. Going beyond PageRank and HITS, we would like to measure the effectiveness of other link-based ranking algorithms, such as SALSA. Finally, we are planning to experiment with more complex feature combinations.

9. REFERENCES

[1] B. Amento, L. Terveen, and W. Hill. Does authority mean quality? Predicting expert quality ratings of web documents. In *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 296–303, 2000.

[2] M. Bianchini, M. Gori, and F. Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 5(1):92–128, 2005.

[3] A. Borodin, G. O. Roberts, and J. S. Rosenthal. Finding authorities and hubs from link structures on the World Wide Web. In *Proc. of the 10th International World Wide Web Conference*, pages 415–429, 2001.

[4] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Transactions on Interet Technology*, 5(1):231–297, 2005.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[6] C. Burges, T. Shaked, E. Renshaw, A. Lazier,

M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. of the 22nd International Conference on Machine Learning*, pages 89–96, New York, NY, USA, 2005. ACM Press.

[7] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *Proc. of the 17th International Conference on Machine Learning*, pages 167–174, 2000.

[8] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 416–423, 2005.

[9] E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178(4060):471–479, 1972.

[10] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *1st International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[11] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proc. of the 30th International Conference on Very Large Databases*, pages 576–587, 2004.

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

[13] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[14] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Extrapolation methods for accelerating PageRank computations. In *Proc. of the 12th International World Wide Web Conference*, pages 261–270, 2003.

[15] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25, 1963.

[16] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[17] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[18] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):2005, 335-380.

[19] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks and ISDN Systems*, 33(1–6):387–401, 2000.

[20] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 258–266, 2001.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[22] J. A. Tomlin. A new paradigm for ranking pages on the World Wide Web. In *Proc. of the 12th International World Wide Web Conference*, pages 350–355, 2003.

[23] T. Upstill, N. Craswell, and D. Hawking. Predicting fame and fortune: Pagerank or indegree? In *Proc. of the Australasian Document Computing Symposium*, pages 31–40, 2003.

[24] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC–13: Web and HARD tracks. In *Proc. of the 13th Text Retrieval Conference*, 2004.