Comparing the Effectiveness of HITS and SALSA

Marc Najork
Microsoft Research
1065 La Avenida
Mountain View, CA 94043, USA
najork@microsoft.com

ABSTRACT

This paper compares the effectiveness of two well-known query-dependent link-based ranking algorithms, "Hyperlink-Induced Topic Search" (HITS) and the "Stochastic Approach for Link-Structure Analysis" (SALSA). The two algorithms are evaluated on a very large web graph induced by 463 million crawled web pages and a set of 28,043 queries and 485,656 results labeled by human judges. We employed three different performance measures – mean average precision (MAP), mean reciprocal rank (MRR), and normalized discounted cumulative gain (NDCG). We found that as an isolated feature, SALSA substantially outperforms HITS. This is quite surprising, given that the two algorithms operate over the same neighborhood graph induced by the query result set. We also studied the combination of SALSA and HITS with BM25F, a state-of-the-art text-based scoring function that incorporates anchor text. We found that the combination of SALSA and BM25F outperforms the combination of HITS and BM25F. Finally, we broke down our query set by query specificity, and found that SALSA (and to a lesser extent HITS) is most effective for general 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

HITS, SALSA, link-based ranking, retrieval performance, web search

CIKM'07, November 6–8, 2007, Lisboa, Portugal.

1. INTRODUCTION

One of the fundamental problems in Information Retrieval is the ranking of search results. In the context of web search, where the corpus is massive and queries rarely contain more than three terms, most searches produce hundreds of results. Given that the majority of search engine users examine only the first page of results [7], effective ranking algorithms are key to satisfying users' needs.

Leading search engines rely on many features in their ranking algorithms. Sources of evidence can include textual similarity between query and documents (or query and anchor texts of hyperlinks pointing to documents), the popularity of documents with users (measured for instance via browser toolbars or by clicks on links in search result pages), and finally hyperlinkage between web pages, which is viewed as a form of peer endorsement among content providers.

Over the past decade, there has been an abundance of research on link-based ranking algorithms. Most of this research has centered around proposing new link-based ranking algorithms or improving the computational efficiency of existing ones (primarily PageRank), but there are only very few published studies on validating the effectiveness (ranking performance) of well-known algorithms on real and large-scale data sets. We believe that this is primarily due to the fact that conducting such studies requires substantial resources: large web graphs, which are typically obtained by crawling a substantial portion of the web; query logs, which are hard to obtain from commercial search engines due to privacy concerns; and human relevance judgments of result sets, which are expensive to produce.

This paper presents a follow-on study on earlier work [12], where we compared the performance of arguably the two most famous link-based ranking algorithms, PageRank [13] and HITS [6], with a state-of-the-art text-based scoring function (BM25F [15]) and what we considered the base-line link-based feature (web page in-degree). To our great surprise, this earlier study found that a base-line link-based feature (the in-degree of web pages, considering only hyperlinks from web pages in a different domain) outperformed both HITS and PageRank, and that the text-based scoring function BM25F vastly outperformed all of the link-based features. Moreover, we found that HITS performed slightly worse than in-degree, despite the fact that it is a query-dependent feature (and relatively expensive to compute).

One might conclude that the effectiveness of link-based features in the ranking of web search results has been overstated all along. Alternatively, one might take the view that link-based features, since they signify peer endorse-

ment of content providers, have gradually deteriorated over the past decade, due to the fact that the web has morphed from being a largely non-commercial space in 1992 to being overwhelmingly commercial today, making objective endorsements among competing content providers much less likely. This decrease in the fraction of non-nepotistic hyperlinks is further aggravated by the fact that most users navigate to web sites through the mediation of a search engine or one of a handful of popular portals, and consequently very few private individuals feel compelled to publish their collection of bookmarks — a practice that was quite common in the early days of the web, and undoubtedly provided a fair amount of unbiased evidence.

However, the above conclusion turns out to be at least partly unwarranted. This paper shows that SALSA [9, 10], a query-dependent link-based ranking algorithm inspired by HITS and PageRank, is substantially more effective as an isolated feature than any of the link-based features examined in our earlier paper, although it still falls well short of the state-of-the-art textual scoring function BM25F. When combining any of the link-based features with BM25F, SALSA's advantage mostly disappears, but it is still the best link-based feature. Finally, when breaking down our query set by query specificity, we found that SALSA is particularly effective for very general queries.

The study described in this paper was conducted on the same data sets (the same web graph and the same set of queries and labeled results) as our earlier study, and uses the same measures of retrieval performance (mean average precision, mean reciprocal rank, and normalized discounted cumulative gain), allowing for an "apples to apples" comparison.

The remainder of this paper is structured as follows: section 2 surveys related work; section 3 characterizes our data sets; section 4 reviews the measures we used to assess ranking performance; section 5 describes the HITS and SALSA algorithms; section 6 gives a short overview of the Scalable Hyperlink Store, the computational infrastructure we used to implement HITS and SALSA; section 7 presents our experimental results; and section 8 offers concluding remarks and avenues for future research.

2. RELATED WORK

The idea of using peer endorsement between web content providers, manifested by hyperlinks between web pages, as evidence in ranking dates back to the mid-1990's. Within a span of 12 months, Marchiori proposed considering links as endorsements [11], Kleinberg introduced HITS, an algorithm that computes hub and authority scores for pages in the distance-one neighborhood of the result set, and Page *et al.* described PageRank [13], an algorithm that computes the global importance of a web page and that was intended to improve on Marchiori's simple link-counting by recursively taking the importance of endorsing pages into account. Both HITS and PageRank proved to be highly influential algorithms in the web research community, inspiring a large amount of follow-on work. A particular interesting such instance is Lempel and Moran's SALSA algorithm [9], which combines key ideas from HITS and PageRank. Lempel and Moran's paper presents persuasive arguments for the merits of SALSA and provides a thorough analysis of its mathematical properties, but their experimental validation was fairly weak (presumably for the very reasons stated in section 1): They conducted their evaluation on a mere five queries, using the AltaVista search engine to compile result sets for each query and AltaVista's link: feature to obtain back-link information. AltaVista's link: feature returned some back-links, but by no means all and certainly not a uniform random sample.

This paper is follow-on work to our earlier study of the relative performance of HITS, PageRank, web page in-degree, and BM25F [12]. It is based on the same data sets, uses the same retrieval performance measures, and compares our new measurements of the performance of SALSA with the measurements presented in that earlier paper.

We are aware of one earlier study that tried to assess the performance of SALSA and compared it to that of HITS, PageRank and in-degree. That study by Borodin *et al.* was based on 34 queries, result sets of 200 pages per query obtained from Google, and a neighborhood graph derived by retrieving 50 back-links per result using Google's link: feature, which has the same limitations as AltaVista's link: feature. By contrast, our study is conducted on a set of over 28,000 queries and a web graph containing close to 3 billion URLs.

3. OUR DATA SETS

The study presented in this paper is based on the same two data sets used in our earlier comparison of HITS with PageRank and in-degree [12]. These two data sets are a large web graph and a substantial set of queries with associated results, some of which were labeled by human judges.

The web graph was obtained by performing a breadth-first search web crawl that retrieved 463,685,607 pages. These pages contain 17,672,011,890 hyperlinks (after eliminating duplicate links embedded in the same web page), which refer to a total of 2,897,671,002 distinct URLs. The mean out-degree of a crawled web page is 38.11; the mean in-degree of discovered pages (whether crawled or not) is 6.10.

Our query set was produced by sampling 28,043 queries from the Live Search query log, and retrieving a total of 66,846,214 result URLs for these queries, or about 2,838 results per query on average. It should be pointed out that our web graph covers only 9,525,566 pages or 14.25% of the result set. 485,656 of the results in the query set (about 17.3 results per query) were rated by human judges as to their relevance to the given query using a six point scale, the ratings 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 biased towards documents considered relevant by pre-existing ranking algorithms. Our performance measures (described in the following section) treat unlabeled results as "detrimental". Spot-checking the set of unlabeled results suggests that this assumption is indeed reasonable.

4. MEASURES OF EFFECTIVENESS

The study described in this paper used the same retrieval performance measures we employed in our earlier comparison of HITS, PageRank and in-degree: Mean average precision, mean reciprocal rank, and normalized discounted cumulative gain. In this section, we will briefly review their respective definitions. In the following, given a rank-ordered vector of n results, let *rat(i)* be the rating of the result at

rank i, with 5 being "definitive" and 0 being "detrimental" or "unlabeled", and let $rel(i)$ be 1 if the result at rank i is relevant[1] and 0 otherwise.

4.1 Mean Average Precision

The *precision* $P@k$ at document cut-off value k is defined to be $\frac{1}{k}\sum_{i=1}^{k} rel(i)$, *i.e.* the fraction of relevant results among the k highest-ranking results. The *average precision* at document cut-off value k is defined to be:

$$AP@k = \frac{\sum_{i=1}^{k} rel(i)\, P@i}{\sum_{i=1}^{n} rel(i)}$$

The *mean average precision* $MAP@k$ at document cut-off value k of a query set is the (arithmetic) mean of the average precisions of all queries in the query set.

4.2 Mean Reciprocal Rank

The *reciprocal rank* at document cut-off value k is defined to be:

$$RR@k = \begin{cases} \frac{1}{i} & \text{if } \exists i \le k : rel(i) = 1 \wedge \forall j < i : rel(j) = 0 \\ 0 & \text{otherwise} \end{cases}$$

The *mean reciprocal rank* $MRR@k$ at document cut-off value k of a query set is the mean of the reciprocal ranks of all queries in the query set.

4.3 Normalized Discounted Cumulative Gain

The *normalized discounted cumulative gain* measure [8] is a non-binary, graded measure that considers all documents in the result set, but discounts the contribution of low-ranking documents. NDCG is actually a family of performance measures. In this study, we used the following instantiation: We define the *discounted cumulative gain* at document cut-off value k to be:

$$DCG@k = \sum_{i=1}^{k} \frac{1}{\log(1+i)} \left(2^{rat(i)} - 1\right)$$

The *normalized discounted cumulative gain* $NDCG@k$ of a scored result set is defined to be the $DCG@k$ of the result set rank-ordered according to the scores divided by the $DCG@k$ of the result set rank-ordered by an "ideal" scoring function, one that rank-orders results according to their rating.

5. HITS AND SALSA

In the mid-1990s, Jon Kleinberg proposed an algorithm called *Hypertext-Induced Topic Search* or HITS for short [6]. HITS is a query-dependent algorithm: It views the documents in the result set as a set of nodes in the web graph; it adds some nodes in the immediate neighborhood in the graph to form a *base set*, it projects the base set onto the full web graph to form a neighborhood graph, and finally it computes two scores, a *hub* score and an *authority* score, for each node in the neighborhood graph.

By contrast, the PageRank algorithm computes the query-independent *importance* of a web page [13]. A web page u with importance score ("PageRank") $R(u)$ propagates a uniform fraction of its score to each of the pages it links to. For technical reasons, the propagation of scores is attenuated by a damping factor, and each node in the web graph receives a share of the scores that are thus diverted. PageRank is often viewed as a random walk over the web graph, and in that view the score of a page is the stationary probability that a node is currently being visited by the random process.

The *Stochastic Approach to Link-Sensitivity Analysis* (or SALSA for short) combines key ideas from HITS and PageRank. SALSA uses exactly the same definition of query-specific neighborhood graph as HITS does, and it also computes a hub score and an authority score for each node in the neighborhood graph. However, while HITS uses an approach called "mutual enforcement" where hubs enforce authorities and vice versa, SALSA computes these scores by performing two independent random walks on the neighborhood graph, a *hub walk* and an *authority walk*, thus adopting a key idea of PageRank.

The remainder of this section provides a formal definition of neighborhood graph, using the same notation as our earlier paper [12], and then describes the HITS and SALSA algorithms.

5.1 The neighborhood graph of a result set

HITS and SALSA are based on 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 page with high in-degree 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, we 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 $|A| \le n$.

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.

[1] In this study, we consider a result to be relevant if it has a label of "good" or better, and irrelevant if it has a label of "fair" or worse". We did investigate if considering documents labeled "fair" as relevant would lead to any qualitative change in results, and found that not to be the case.

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.

5.2 The HITS algorithm

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:

HITS-Hub-and-Authority-Scores:

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) $H := \frac{1}{\|H'\|_2} H', A := \frac{1}{\|A'\|_2} A'$

where $\|X\|_2$ is the euclidean norm of vector X.

5.3 The SALSA algorithm

For each node u in the neighborhood graph, SALSA computes an *authority score* $A(u)$ and a *hub score* $H(u)$ using the following two *independent* algorithms:

SALSA-Hub-Scores:

1. Let B^H be $\{u \in B : out(u) > 0\}$.
2. For all $u \in B$:
$$H(u) := \begin{cases} \frac{1}{|B^H|} & \text{if } u \in B^H \\ 0 & \text{otherwise} \end{cases}$$
3. Repeat until H converges:
   (a) For all $u \in B^H$:
$$H'(u) := \sum_{(u,v) \in N} \sum_{(w,v) \in N} \frac{H(w)}{in(v)out(w)}$$
   (b) For all $u \in B^H : H(u) := H'(u)$

SALSA-Authority-Scores:

1. Let B^A be $\{u \in B : in(u) > 0\}$.
2. For all $u \in B$:
$$A(u) := \begin{cases} \frac{1}{|B^A|} & \text{if } u \in B^A \\ 0 & \text{otherwise} \end{cases}$$
3. Repeat until A converges:
   (a) For all $u \in B^A$:
$$A'(u) := \sum_{(v,u) \in N} \sum_{(v,w) \in N} \frac{A(w)}{out(v)in(w)}$$
   (b) For all $u \in B^A : A(u) := A'(u)$

6. THE SCALABLE HYPERLINK STORE

We implemented HITS and SALSA on top of the *Scalable Hyperlink Store*, a special-purpose storage system for the web graph. SHS was heavily influenced by the Compaq Link Database [14], but unlike that system, SHS is distributed over many machines. It maintains the web graph in main memory to allow extremely fast random access to nodes (URLs) and edges (hyperlinks), and it uses data compression techniques that leverage structural properties (namely, the prevalence of relative links) of the web graph to achieve fairly good compression. Serving the full 17.7 billion link graph mentioned in section 3 requires six machines, each with 16 GB of main memory.

The two principal abstractions used in SHS are a *URL store* and two *link stores*, one to trace links forward and another to trace them back. Clients use SHS by linking against a library containing classes that implement clerks for the URL store and the link stores; all the intricacies common to distributed systems are handled by the clerks and the SHS servers.

The URL store maintains a bijection between URLs (strings) and UIDs (integers that serve as short-hands for URLs). Clients can map URLs to UIDs and UIDs back to URLs. The API of the URL store clerk looks as follows:

```
class UrlStoreClerk {
  ... // omitting private members
 public:
  UrlStoreClerk(char *serverNameFile);
  ~UrlStoreClerk();
  INT64 UrlToUid(char *url);
  char *UidToUrl(INT64 uid);
  SeqInt64 BatchedUrlToUid(SeqString& urls);
  SeqString BatchedUidToUrl(SeqInt64& uids);
  ... // omitting methods irrelevant to this paper
};
```

The *UrlStoreClerk* constructor takes the name of a file that contains the names of the SHS servers maintaining the graph. The central methods are *UrlToUid*, which maps a URL to a UID, and *UidToUrl*, which maps a UID back to a URL. The methods *BatchedUrlToUid* and *BatchedUidToUrl* are variants of the previous two methods that allow the mapping of entire batches of URLs or UIDs; their purpose is to allow client applications to amortize RPC overheads. As a point of reference, mapping a URL to a UID takes about 3 microseconds, while performing a null RPC takes about 100 microseconds; so providing a mechanism to batch up requests is performance-critical. Our implementations of

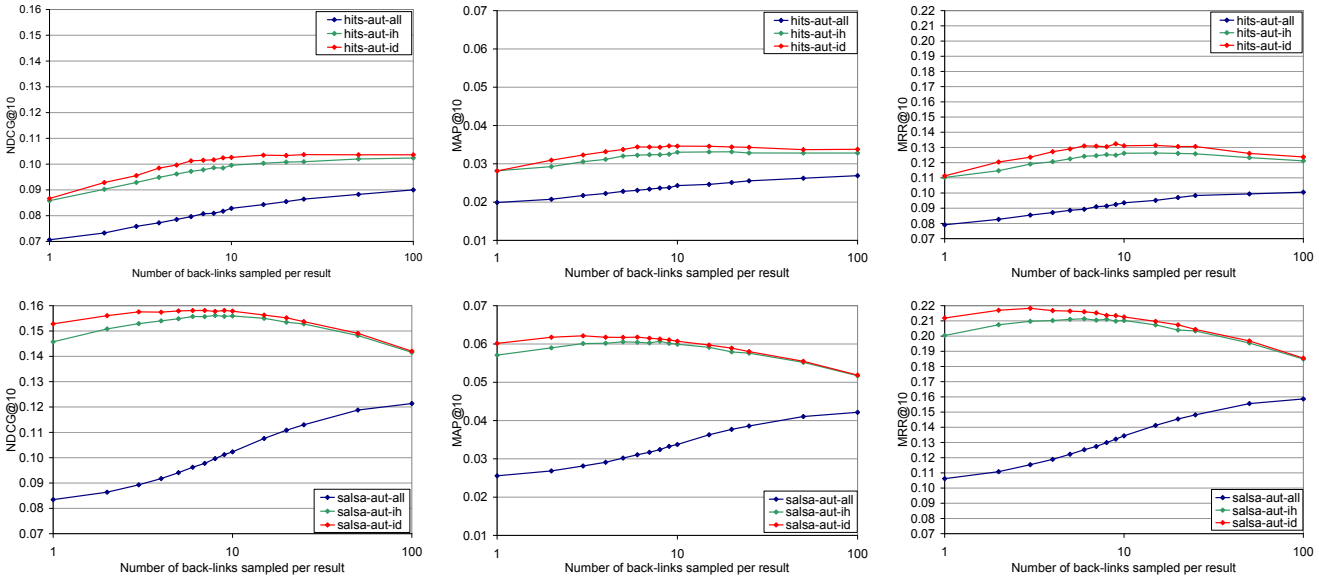

Figure 1: Effectiveness of authority scores computed using different parameterizations of HITS and SALSA; measured in terms NDCG, MAP and MRR.

HITS and SALSA perform a single call to *BatchedUrlToUid* per query.

An SHS service maintains two link stores: One for determining the outgoing links of a given web page, and one for determining the incoming links. The API of the link store clerk is as follows:

```
class LinkStoreClerk {
  ... // omitting private members
 public:
  LinkStoreClerk(char *serverNameFile, bool fwdDB);
  ~LinkStoreClerk();
  SeqInt64 GetLinks(INT64 uid);
  SeqInt64 SampleLinks(INT64 uid, int num);
  SeqInt64 *BatchedGetLinks(SeqInt64& uids);
  SeqInt64 *BatchedSampleLinks(SeqInt64& uids, int num);
  ... // omitting methods irrelevant to this paper
};
```

The *LinkStoreClerk* constructor takes the name of a file that contains the names of the SHS servers maintaining the graph, and a boolean value indicating whether to access the forward or the backward link store. The method *GetLinks* takes a UID u and returns the set of UIDs that u links to (or that link to u, if the backward store is consulted). The method *SampleLinks* takes a UID u and an integer n, and returns a uniform random sample of n UIDs that u links to (or that link to u); if there are fewer than n such UIDs, all are returned. The methods *BatchedGetLinks* and *BatchedSampleLinks* are variants of the previous two methods that allow client applications to batch up many UIDs, so as to amortize RPC overhead. As a point of reference, *GetLinks* takes about 0.4 microseconds per returned UID, excluding the RPC overhead. Our implementations of HITS and SALSA perform one call to *BatchedSampleLinks* and two calls to *BatchedGetLinks* per query.

7. EXPERIMENTAL RESULTS

For the experiments described in this paper, 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*). Using each of these databases, we computed HITS and SALSA 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 45 HITS and 45 SALSA 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 authority and hub scores, one query at a time. The longest-running combination (SALSA using the *all* database and sampling 100 back-links of each root set vertex) required 60,425 seconds to process the entire query set, or about 2.15 seconds per query on average.

The first question we are interested in concerns the relationship between the performance of HITS and SALSA and the number of back-links sampled per result. One would expect that sampling more back-links should improve the effectiveness of both HITS and SALSA, since it leads to a closer approximation of the complete neighborhood graph. However, we were surprised! Figure 1 shows the retrieval performance of HITS and SALSA authority scores as a function of the number of sampled back-links. The figure contains six graphs, for the two algorithms (HITS and SALSA) times the three performance measures used (NDCG, MAP, and MRR, all at document cut-off value 10). Each graph shows three curves, one for each of the three SHS databases (*all*, *ih*, and *id*). The horizontal axes (drawn on a log scale) denote the number of back-links sampled per result, the vertical axes denote retrieval performance.

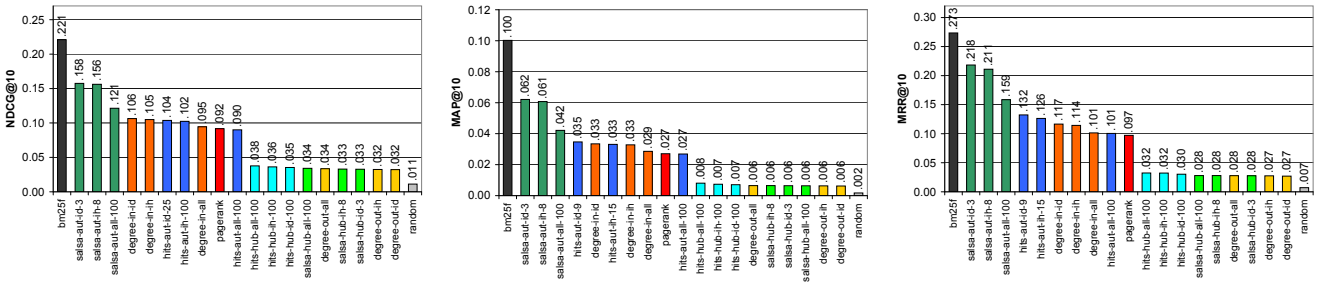

Figure 2: Effectiveness of different isolated features; measured in terms of NDCG, MAP and MRR.

The *all* versions of both HITS and SALSA behave according to our intuition. The performance of each variant increases as more back-links are sampled. It also appears that at beyond ten samples, the slope of each curve is decreasing, suggesting that there are diminishing returns to including more samples.

The *ih* and *id* versions of HITS show a mixed behavior. According to the NDCG measure, performance increases as more back-links are sampled, much in the same way as it does for the *all* version; however, the absolute performance is substantially higher. According to both MAP and MRR, performance maximizes at about 10 samples, and levels out slightly below the maximum when more samples are drawn.

Finally, the *ih* and *id* versions of SALSA behave differently from what our intuition would suggest. Depending on the measure used, performance is maximal for between 3 and 8 samples, and falls off in a pronounced fashion as more samples are drawn. This behavior persists across the entire possible range of document cut-off values (not shown for reasons of space).[2] At this point, we have no clear explanation for this counterintuitive behavior. Our best hypothesis is that a small fraction of the nodes in I_∞^{ih} or I_∞^{id} have a deleterious effect on effectiveness, and that they are more likely to be included in I_s^{ih} or I_s^{id} as s is increased. Testing this hypothesis is our next goal.

Next, we compare the effectiveness of SALSA as an isolated feature to that of other features. Figure 2 compares SALSA to all the features covered in our earlier paper [12]: HITS, PageRank, web page in- and out-degree, and BM25F. The figure shows three graphs, one for each of the three performance measures. The vertical axis denotes retrieval performance. Each graph shows a set of bars, one per isolated feature; the bars are ordered by decreasing height. Under all measures, the three variants of SALSA authority scores (with the number of sampled back-links chosen to maximize effectiveness) clearly outperform all other link-based features, although they are still well below the performance of BM25F. The *ih* and *id* variant of SALSA authority scores perform about equally well, while the *all* variant fares substantially worse. The performance of SALSA hub scores, on the other hand, is indistinguishable from that of the other out-link based features.

Actual retrieval systems rely on multiple sources of evidence (hundreds in the case of commercial search engines), and combine evidence in various ways. In order to assess the impact of a single feature on the overall performance of a scoring function, it is not enough to measure the performance of the feature in isolation, since it may be correlated with other features that are provided to the scoring function. One must measure the impact of including this feature on the performance of the combined evidence.

We chose a fairly simple model for combining features: Given a set of n features $F_i(d)$ (with $1 \leq i \leq n$) of a result document d, we apply a feature-specific transform T_i, adjust the contribution of the transformed feature using a scalar weight w_i, and add up the contributions of the individual features. This leads us with the following scoring function:

$$score(d) = \sum_{i=1}^{n} w_i T_i(F_i(d))$$

For each feature, we chose a transform function that we empirically determined to be well-suited. Table 1 shows the chosen transform functions. We tuned the scalar weights by selecting 5000 queries at random from the test set, using an iterative refinement process to determine the weight that maximized the given performance measure, fixed the weight, and used the remaining 23,043 queries to assess the performance of the scoring function.

We combined each of the link-based features with BM25F, our state-of-the-art textual feature, using the above scoring function, and measured the performance. Figure 3 shows the results, using the same visualization as figure 2; however, note that the vertical axes do not start at 0. The right-most bar in each graph shows the performance of BM25F as an isolated feature, so as to provide a baseline.

According to the MAP and MRR measures, the combination of BM25F and SALSA authority scores performs best, although the margin of the combination of BM25F and webpage in-degree or PageRank is rather slim. According to the NDCG measure, SALSA authority scores and page in-

Feature	Transform function
salsa-aut-*	$T(s) = \log(s + 3 \cdot 10^{-6})$
salsa-hub-*	$T(s) = \log(s + 3 \cdot 10^{-2})$
hits-aut-*	$T(s) = \log(s + 3 \cdot 10^{-3})$
hits-hub-*	$T(s) = \log(s + 1 \cdot 10^{-1})$
degree-in-*	$T(s) = \log(s + 3 \cdot 10^{-2})$
degree-out-*	$T(s) = \log(s + 3 \cdot 10^{3})$
pagerank	$T(s) = \log(s + 3 \cdot 10^{-12})$
bm25f	$T(s) = s$

Table 1: Near-optimal feature transform functions.

[2]This anomaly manifests itself in a similar or even more pronounced fashion for other SALSA-like algorithms that we experimented with, but that are beyond the scope of this paper.

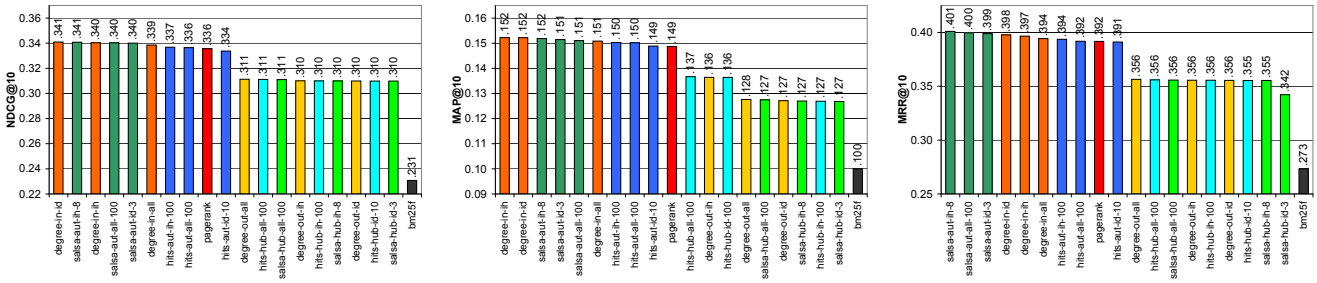

Figure 3: Effectiveness measures for linear combinations of link-based features with BM25F.

degree are about tied. Overall, any feature dominated by incoming links (page in-degree, HITS and SALSA authority scores, and PageRank) seems to improve performance by roughly the same amount over the BM25F baseline. We speculate that this is due to the fact that these features are not actually independent: BM25F itself incorporates in-link information, in the guise of anchor text.

We should stress that the results shown in figure 3 are not necessarily damning for sophisticated (and accordingly computationally expensive) link-based ranking algorithms. It might well be that the various link-based features would exhibit a more differentiated behavior if they were combined with BM25F using a more sophisticated scoring function, such as a two-layered RankNet [4].

Finally, we investigated the relationship between "query specificity" and isolated features. Ideally, we would quantify the specificity of a query by the cardinality of the result set it produces. General queries produce large result set (making good ranking algorithms all the more important), whereas specific queries produce smaller result sets. Unfortunately, our query set does not contain the size of the result set. Therefore, we adopted the same approach as in our earlier paper, and approximated query specificity by the sum of the inverse document frequencies of the individual query terms. Recall that the IDF of a term t with respect to a document collection C is defined to be $\log \frac{|C|}{|C(t)|}$, where $C(t)$ is the subset of documents in C containing t. By summing up IDFs of the individual query terms, we make the (unwarranted) assumption that the terms in a single query are independent of each other. This approximation will overestimate the specificity of a query. Alas, while not perfect, it is at least directionally accurate.

We broke our query set down into 13 subsets according to specificity, and used each of five selected features (PageRank, *id* in-degree, HITS authority scores computed on the B^{id}_{100} neighborhood, SALSA authority scores computed on the B^{id}_{3} neighborhood, and BM25F) in isolation to rank the queries in each subset. Figure 4 shows the performance of each feature for each query subset. As usual, the figure shows three graphs, one per performance measure. The lower horizonal axis of each graph shows query specificity (the most general queries being on the far left); the upper horizontal axis shows the size of each of the 13 query subsets. The vertical axis denotes retrieval performance. Each graph contains five curves, one for each of the chosen features. We can see that all the link-based features perform best for fairly general queries (peaking at an IDF sum of 4 to 8), whereas BM25F performs best for moderately specific queries (peaking at an IDF sum of 10 to 14). Among the link-based features, SALSA authority scores clearly perform best, dominating all other link-based features across the entire query specificity range. According to the MAP and MRR measures, none of the five features performs very well for highly specific queries, although BM25F outperforms all the link-based features, as is to be expected for highly discriminative queries. The NDCG graph has outliers for query specificities $[20, 22)$ and $[24, \infty)$, which we don't fully understand but attribute to noise.

Figure 4 suggests that web search engines should weigh evidence differently when scoring the results to a query depending on its specificity. It is reasonable to assume that we could break down queries along other dimensions as well (say, navigational vs. transactional), and see a similarly differentiated behavior.

8. CONCLUSIONS AND FUTURE WORK

This paper describes a large-scale evaluation of the performance of SALSA relative to other link-based features. It builds on an earlier comparison of HITS, PageRank, in-degree, and BM25F. While our earlier study found that HITS and PageRank were underperforming the base-line link-based feature of inter-domain in-degree, casting doubt on the usefulness of sophisticated link-based features in the ranking of web search results, this study finds that SALSA, a query-dependent link-based ranking algorithm, substantially outperforms the link-based features examined in our earlier study. It also finds that SALSA is particularly effective at scoring fairly general queries.

We hope (and have some reason to believe) that there exist other query-dependent link-based ranking algorithms that perform yet better. Our next goal is to investigate the "sampling anomaly": one would expect that increasing the number of result set ancestors sampled into the neighborhood graph would more closely approximate the structure of the full distance-one neighborhood graph, and should thus lead to better retrieval performance. However, this is not true for SALSA (and incidentally also not true for other algorithms that we have experimented with but that are beyond the scope of this paper). We believe that understanding the cause of this anomaly will provide us with a deeper understanding of how some nodes in the neighborhood graph can have a deleterious effect on ranking algorithms, and hopefully lead to heuristics for excluding such nodes from the ranking computation and thus increasing ranking performance.

Going beyond that, we are planning to experiment with variations of HITS and SALSA, and with other query-dependent ranking algorithms proposed in the literature, such as

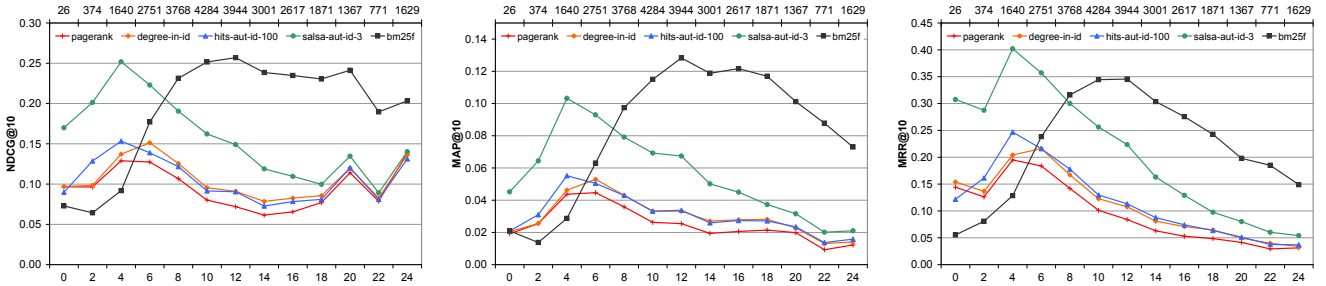

Figure 4: Effectiveness measures for selected isolated features, broken down by query specificity.

the MAX algorithm proposed by Borodin *et al.* We feel confident that link-based ranking features still hold great (and untapped) potential.

9. ACKNOWLEDGEMENTS

I would like to thank Hugo Zaragoza for introducing me to IR performance evaluation, Hugo and Mike Taylor for collaborating on an earlier study and for shaping my thinking, Nick Craswell for clarifying my understanding of HITS and SALSA, Frank McSherry for tackling the issue of tied scores, and Michael Isard for numerous helpful discussions.

10. 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] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] R. Lempel and S. Moran. SALSA: The stochastic approach for link-structure analysis. *ACM Transactions on Information Systems*, 19(2):131–160, 2001.

[11] M. Marchiori. The quest for correct information on the Web: Hyper search engines. In *Computer Networks and ISDN Systems*, 29(8–13):1225–1236, 1997.

[12] M. Najork, H. Zaragoza and M. Taylor. HITS on the Web: How does it Compare? In *30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 471–478, 2007.

[13] 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.

[14] K. Randall, R. Stata, J. Wiener and R. Wickremesinghe. The Link Database: Fast Access to Graphs of the Web. In *Proc. of the Data Compression Conference*, pages 122–131, 2002.

[15] 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.