Measuring Similarity to Detect Qualified Links*

Xiaoguang Qi, Lan Nie and Brian D. Davison
Department of Computer Science & Engineering
Lehigh University
{xiq204,lan2,davison}@cse.lehigh.edu

December 2006

Abstract

The success of link-based ranking algorithms is achieved based on the assumption that links imply merit of the target pages. However, on the real web, there exist links for purposes other than to confer authority. Such links bring noise into link analysis and harm the quality of retrieval. In order to provide high quality search results, it is important to detect them and reduce their influence. In this paper, a method is proposed to detect such links by considering multiple similarity measures over the source pages and target pages. With the help of a classifier, these noisy links are detected and dropped. After that, link analysis algorithms are performed on the reduced link graph. The usefulness of a number of features are also tested. Experiments across 53 query-specific datasets show that the result of our approach is able to boost Bharat and Henzinger's *imp* algorithm by around 9% in terms of precision. It also outperforms a previous approach focusing on link spam detection.

1 Introduction

In modern web search engines, link-based ranking algorithms play an important role. Typical link analysis algorithms are based on the assumption that links confer authority. However, this assumption is often broken on the real web. As a result, the retrieval performance based on such naive link analysis is often disappointing. According to our preliminary experiment on more than fifty query-specific datasets, on average, only four out of the top ten results generated by the HITS algorithm [9] are considered relevant to the query by the users (details in Section 5).

The prevalence of links that do not (or should not) confer authority is an important reason that makes link analysis less effective. Examples of such links are links that are created for the purpose of advertising or navigation. Figure

*Technical Report LU-CSE-06-033, Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, 18015.

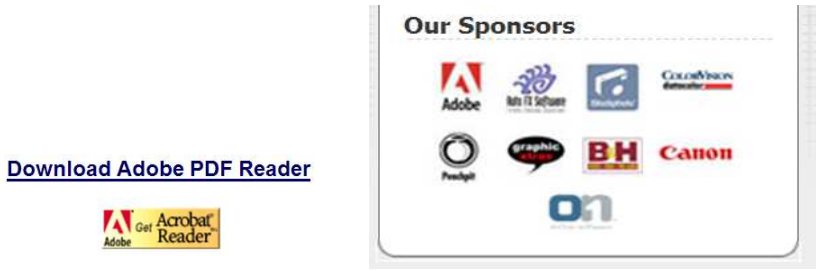

Figure 1: Examples of links that do not confer authority

1 shows some of such links. This type of links are very common on the Web. From a person's view, these links do carry some information that the authors of the web pages want to spread. However, from the perspective of link analysis algorithms, these links are noisy information because they do not show the authors' recommendation of the target pages. Traditional link analysis algorithms do not distinguish such noise from useful information. As a consequence, the target pages of these links could get unmerited higher ranking. Therefore, in order to provide better retrieval quality, the influence of such links needs to be reduced.

For years, researchers have been working on improving the quality of link analysis ranking. Advanced algorithms are proposed based on traditional PageRank [12] and HITS [9]. Some other work focuses on detecting and demoting web pages that do not deserve the ranking generated by traditional link analysis. However, little work has been done in the aspect of filtering out noisy information (nonuseful links) and preventing them from being used by link analysis.

In this paper, we introduce the notion of "qualified links"— links that are qualified to make a recommendation regarding the target page. We propose to detect qualified links using a classifier which, based on a number of similarity measures of the source page and target page of a link, makes the decision that whether the link is "qualified". After this, the "unqualified links" are filtered out, which leaves only the "qualified links". Link analysis algorithms are then performed on the reduced web graph and generate the resulting authority ranking. We also studied a number of features in the "qualified link classification", revealing some interesting insights.

The contributions of this paper are:

- the novel notion of "qualified links" and a method to differentiate such links from those "unqualified";
- a study of the features being used to detect "unqualified links";
- an experimental comparison of our approach with other web ranking algorithms on real-world datasets.

The rest of this paper is organized as follows. The background of link analysis and related work in link spam detection and demotion is briefly reviewed in Section 2. Our motivation is presented in Section 3 and the methodology is detailed in Section 4. In Section 5, experimental results are presented. Finally, we conclude this paper with a discussion.

2 Background and related work

The idea of incorporating link analysis in ranking algorithms was first considered almost a decade ago. In this section, we briefly review the background of link-based ranking algorithms and the related work in link spam detection.

2.1 Background

2.1.1 Hyperlink-Induced Topic Search (HITS)

While at IBM Research, Jon Kleinberg proposed [9] that web documents had two important properties, called hubness and authority, as well as a mechanism to calculate them. Pages functioning as good hubs have links pointing to many good authority pages, and good authorities are pages to which many good hubs point. Thus, in his Hyperlink-Induced Topic Search (HITS) approach to broad topic information discovery, the score of a hub (authority) depended on the sum of the scores of the connected authorities (hubs):

$$A(p) = \sum_{q:q \to p} H(q) \text{ and } H(p) = \sum_{q:p \to q} A(q)$$

Kleinberg didn't calculate these scores globally; instead, he used the subset of the web that included top-ranked pages for a given query, plus those pages pointed to and were pointed by that set, the union of which he called the base set.

When first introduced, HITS was able to work with the existing search engines (which were mostly textually based) and generate results (at least for broad topics) that were comparable to hand-compiled information in directories such as Yahoo! [14].

HITS uses the results of a search engine query to make its analysis query-specific, but sometimes the topical focus of the base set can drift to a broader or more popular topic. In addition, the one-step expansion can bring in unrelated yet popular pages that can end up with high ranks.

2.1.2 Bharat and Henzinger's improvements to HITS

Bharat and Henzinger [1] proposed a number of improvements to HITS. The first change is an algorithm called *imp*, which re-weights links involved in mutually reinforcing relationships and drops links within the same host. In order to reduce topic drift, they eliminate documents that were not sufficiently similar

to the query topic, which was comprised of the first 1,000 words of each core document. In addition, they used the relevance scores of a node as a weight on its contribution so that the nodes most relevant to the query have the most influence on the calculation. They found that *imp* made a significant improvement over the original HITS.

2.1.3 PageRank

At approximately the same time as Kleinberg, Stanford graduate students Sergey Brin and Lawrence Page proposed an alternative model of page importance, called the random surfer model [12]. In that model, a surfer on a given page i, with probability $(1 - d)$ chooses to select uniformly one of its outlinks $O(i)$, and with probability d to jump to a random page from the entire web W. The PageRank [2] score for node i is defined as the stationary probability of finding the random surfer at node i. One formulation of PageRank is

$$PR(i) = (1 - d) \sum_{j:j \to i} \frac{PR(j)}{O(j)} + d\frac{1}{N}$$

Because the definition of PageRank is recursive, it must be iteratively evaluated until convergence.

PageRank is a topic-independent measure of the importance of a web page, and must be combined with one or more measures of query relevance for ranking the results of a search.

2.2 Related work

Lempel and Moran [10] defined a tightly-knit community (TKC) as a small but highly connected set of sites. Even though such a community is not quite relevant to the query, it may still be ranked highly by link-based ranking algorithms. The authors proposed SALSA, a stochastic approach for link structure analysis, which is less vulnerable to the TKC effect than HITS. This method, however, becomes less effective if spam sites are not densely connected.

Davison [4] proposed the use of decision tree to recognize and eliminate nepotistic links, links that are present for reason other than merit. Drost and Scheffer [5] demonstrated that, trained on a number of manually selected features, classifiers can identify spam pages. These two approaches require a certain amount of human effort to choose features which may need to change over time.

Chakrabarti [3] proposed an approach based on HITS to perform topic distillation on a fine-grained document model. In this model, web pages are represented by their Document Object Model trees. The algorithm disassemble hubs by segmenting the DOM trees. And the mutual reenforcement between hubs and authorities are performed among these disassembled segments.

Li et al. [11] pointed out the small-in-large-out link problem with HITS, in which a community associate with a root with few in-links but many outlinks. Such communities may dominate HITS results even if they are not very

relevant. The authors addressed this problem by assigning appropriate weights to the in-links of root.

Based on the idea that good web sites seldom point to spam sites, Gyöngyi et al. [7] introduced TrustRank to combat web spam. The trust of good sites is propagated through the links on the web. This propagation is calculeted iteratively until convergence, when good sites get higher trust scores, while spam sites get lower scores.

Zhang et al. [15] proposed an algorithm to make eigenvector-based ranking algorithms less vulnerable to noisy information. Fetterly et al. [6] also proposed approaches that can detect spam by statistical analysis.

Wu and Davison [13] proposed a two-step algorithm to identify link farms. The first step generates a seed set based on the size of the intersection of in-link and out-links of web pages. The second step expands the seed set to include the pages pointing to many pages within the seed set. The links between these identified spam pages are then re-weighted and a ranking algorithm is performed on the modified link graph.

3 Motivation

The success of link-based ranking algorithms is achieved based on the assumption that links imply merit of the target pages. However, there are cases in which this assumption does not hold. An evident example is spam links, links that are created for the sole purpose of manipulating the ranking algorithm of a search engine. The presence of link spam makes link analysis less effective. Another example is navigational links, where links are created for easy access to other pages regardless of relevance. Links between different regional web sites of the same company (`http://foobar.com/` and `http://foobar.co.uk/`), and links to the "terms of use" of a web site can be considered as examples of navigational links. Although navigational links are not created for the purpose of spamming, they should also be considered less valuable for link analysis since they hardly imply authority of their target pages.

Based on this motivation, we introduce the notion of "a qualified link". A qualified link is a link on a page that is qualified to make a recommendation regarding the target page, and is in some sense the opposite of a nepotistic link [4].

Besides spam links and navigational links, other types of "unqualified links" include advertising links and irrelevant links. Advertising links are links created for the purpose of advertising. Irrelevant links can be considered as the collection of other "unqualified links", such as links pointing to a required document viewer or to a particular web browser for the desired display of the web page.

To determine whether a link is qualified or not, we propose to build a binary classifier based on the characteristics of links. Then based on the decision of that classifier, a filtering process can be performed on the web graph which filters out the "unqualified" links. Finally, link analysis algorithms can run on the reduced graph and generate rankings.

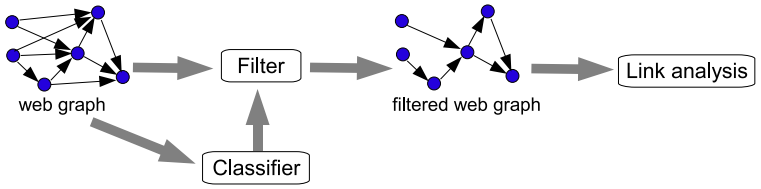

Figure 2: The process of "qualified" link analysis

Alternatively, if the classifier is able to generate a reasonable probability of a link being "qualified", we may consider the probability as the quantitative value of quality. After that, link analysis may be performed on a weighted graph where each edge in the web graph is weighted by its value of quality.

There are a variety of metrics one might use to measure the qualification of a link. The measures we use in this work are the similarity scores of the source page and the target page, such as content similarity and URL similarity. The similarity measures are detailed in Section 4. The classifier considers these similarity measures, and makes a decision (of a link being "qualified" or not) or predict a probability (of how likely a link being "qualified"). The process of this approach is visualized in Figure 2.

4 Qualified link analysis

4.1 Similarity measures

There are a variety of features one might use to measure the qualification of a link. However, considering the issue of computational complexity, it is desirable to use a small number of features and to use features that are easy to compute. We propose predicting a link being "qualified" or not by considering the similarity scores of its source and target pages. Six features are used in this work; they are host similarity, URL similarity, topic vector similarity, tfidf content similarity, tfidf anchor text similarity, and tfidf non-anchor text similarity. The computation of these similarity measures are detailed as follows.

- Host similarity

  The host similarity of two web pages is measured by the portion of common substrings that the host names of those two web pages have. Suppose s is a string and r is an integer, $Substr(s, r)$ is the set of all substrings of s with length r, $host_x$ is the host name of a web page x, then the host similarity of two web pages x and y is calculated by Equation 1.

$$Sim_{host}(x, y) = \frac{|Substr(host_x, r) \cap Substr(host_y, r)|}{|Substr(host_x, r)| + |Substr(host_y, r)|} \quad (1)$$

  In the experiments of this work, r is set to 3.

- URL similarity

  Analogous to host similarity, the URL similarity of two web pages is measured by the common substrings that the URLs of those two web pages have. Still using the notations above and suppose URL_x is the URL of web page x, then the URL similarity of two web pages x and y is calculated by Equation 2.

$$Sim_{URL}(x, y) = \frac{|Substr(URL_x, r) \cap Substr(URL_y, r)|}{|Substr(URL_x, r)| + |Substr(URL_y, r)|} \tag{2}$$

  Here, r is also set to 3.

- Topic vector similarity

  The topic vector similarity reflects how similar the topics of the two web pages are. If there are n pre-defined topics t_1 through t_n, then each web page x can be represented by a probability distribution vector $v_x = (v_{x,1}, v_{x,2}, ..., v_{x,n})$, in which each component $v_{x,i}$ is the probability that page x is on topic t_i. Such a vector can be computed by a textual classifier, such as naive Bayes. The topic vector similarity is computed as the cosine similarity of the topic vectors of the two pages.

$$Sim_{topic}(x, y) = \sum_{i=1}^{n} v_{x,i} \times v_{y,i} \tag{3}$$

- Tfidf content similarity

  The tfidf content similarity of two web pages measures the term-based similarity of their textual content. We use the equations used by the Cornell SMART system to compute the tfidf representation of a web document. Given a collection D, a document $d \in D$, a term t, suppose $n(d, t)$ is the number of times term t occurs in document d, D_t is the set of documents containing term t, then the term frequency of term t in document d is

$$TF(d, t) = \begin{cases} 0 & if\ n(d, t) = 0 \\ 1 + log(1 + log(n(d, t))) & otherwise \end{cases} \tag{4}$$

  The inverse document frequency is

$$IDF(t) = log\frac{1 + |D|}{|D_t|} \tag{5}$$

  In vector space model, each document d is represented by a vector in which each component d_t is its projection on axis t, given by

$$d_t = TF(d, t) \times IDF(t) \tag{6}$$

  Then the content similarity of web pages x and y is computed as the distance of their vector space representations.

$$Sim_{content}(x, y) = \sqrt{\frac{\sum_{t \in T}(x_t - y_t)^2}{\sum_{t \in T} x_t^2 \cdot \sum_{t \in T} y_t^2}} \quad (7)$$

- Anchor text similarity

  The anchor text similarity of two pages measures the similarity of the anchor text in those two pages. It is computed the same way as content similarity, except substituting each document by a "virtual document" consisting of all the anchor text inside that document. Still, the similarity score is computed as the distance of the two vectors, each representing a "virtual document". IDF is estimated on the collection of these "virtual documents".

- Non-Anchor text similarity

  The non-anchor text similarity of two pages measures the similarity of textual content that is not anchor text in those two pages. It is computed the same way as content similarity, except substituting each document by a virtual document consisting of all the textual content inside that document that is not anchor text. IDF is estimated on the collection of the "virtual documents".

4.2 Qualified HITS

HITS uses a two-step process to collect a query-specific dataset. The goal is to produce a small collection of pages likely to contain the most authoritative pages of a given topic. Starting from a given query, HITS assembles an initial collection of pages, typically, up to 200 top ranked pages returned by a text search engine on that query. Although this root set R is rich in relevant documents, it is restricted to those pages containing the query string. For most short queries, especially those representing a broad topic, such a limitation may exclude some of the strong authorities. In addition, there are often extremely few links between pages in R [9], rendering it essentially "structureless" and hard for later link analysis. To solve the problem, an expansion step is evoked from the root set. Consider a relevant page for the query topic, although it may well not be in the set R, it is quite likely to know or to be known by at least one page in R. Hence, the dataset is augmented by adding any pages that are linked to or from a page in the root set R. These interconnected candidates are then analyzed by the HITS algorithm to identify the best authorities.

However, both the dataset collection process and HITS analysis take the "links imply relevancy" for granted. Since they treat all the hyperlinks equally, as analyzed in the above section, they are vulnerable to "unqualified links". Irrelevant pages may dominate the query-specific web graph and ruin the ranking result. For these unqualified hyperlinks, no matter for what reason they are created, they break the relevance assumption, they prevent the dataset from staying on the query topic, and they bring noise to the HITS calculation as

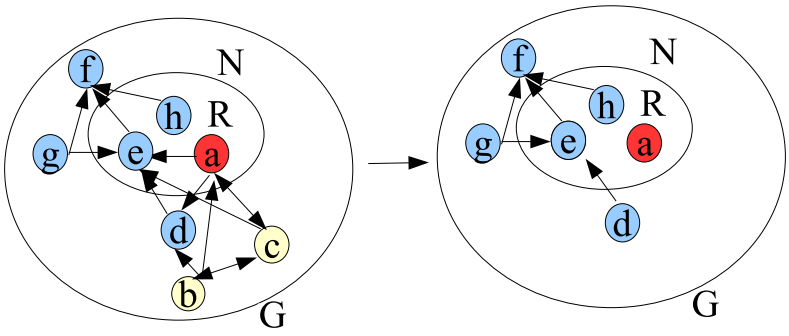

Figure 3: Pruning a query-specific graph.

well. To solve this problem, we propose a simple heuristic approach to eliminate unqualified links and irrelevant pages from the dataset.

Suppose we produce a focused web graph $G(V, E)$ for a given query using the HITS process described above, where V is the set of web pages and E represents hyperlinks among those pages. In addition, V consists of the initial root set R and the set of R's neighboring pages N. We then use the following rules to filter out noises in the graph G. An example is given in Figure 3.

- For every hyperlink in E, compute the similarity scores of its source page and target page. Feed these scores into a classifier which is trained on some labeled links. If the classifier gives a negative answer, we consider this hyperlink is unqualified and should be removed from the graph. In the example, let us suppose links $a \rightarrow d$, $b \rightarrow d$, $b \rightarrow a$, $a \rightarrow c$ and $c \rightarrow a$ are removed.

- Scan the graph with unqualified links eliminated and check each page in the neighboring set N to see if it is still connected with the root set R. If the answer is negative, it is indicated that the page is not relevant to any page in the root set and should not be included in the data set in the beginning. As a result, this page, as well as all the links associated with it, are removed from the link graph. Back to the example, neighboring pages b and c are no longer connected with the root set R and thus are removed, as well as the links between them. d, f and g remain since they are still connected to the root set.

In summary, originally in this example, pages a, b, c and d form a densely-connected community and dominate the link graph. After the two steps above, the graph is converted from the one on the left to the right one in Figure 3. As a result, the connectivity inside this community is reduced and some irrelevant pages are directly removed. The reputation of these pages are thus successfully demoted. On the other hand, those good authorities, such as f and e are not be affected much.

4.3 Qualified PageRank

The method of qualified PageRank is the same as qualified HITS except that the second step is unnecessary since PageRank runs on the global link graph as opposed to a query-specific graph.

5 Experiments

5.1 Datasets

Qualified-HITS needs to be tested on query-specific datasets. In order to evaluate Qualified-HITS, we used the query-specific datasets collected by Wu and Davison[13]. The corpora includes 412 query-specific datasets, with 2.1 million documents. The queries are selected from the queries used in previous research, the category name of ODP directory, and popular queries from Lycos and Google.

The dataset collecting process is similar to the way used in HITS: for each query, they used `search.yahoo.com` to get the top 200 URLs; then for each URL, the top 50 incoming links to this URL are retrieved by querying Yahoo again. All pages referenced by these top 200 URLs are also download.

From this dataset, we randomly selected 58 queries, and used these 58 query-specific datasets to evaluate the performance of Qualified-HITS. A sample of these queries is shown in Table 1.

In their work of spam detection, they presented a two-step algorithm for detecting link farms automatically. As a result, spam pages are identified and the links among them are dropped (or down-weighted).

5.2 Human labeling of links

In order to build a classifier which categorizes links into qualified links and unqualified links, a set of labeled training data is needed. We manually labeled 1247 links that are randomly selected from five query-specific datasets (marked with ** in Table 1). To each link, one of the following labels is assigned: recommendation, navigational, spam, advertising, irrelevant, and undecidable. These labels are not directly used to train the classifier. Instead, they are mapped to two labels, qualified and unqualified. Recommendation links are considered qualified, while, navigational, spam, advertising, and irrelevant links are unqualified. A link is labeled undecidable if the content of its source or target page is not available. This category of links is not used to train the classifier.

Two human editors were involved in this labeling task. In order to estimate how consistent their decisions are, their individual labeling results on 100 links are compared. On 85 links, their decisions are the same. After mapping the labels to qualified or unqualified, they agree on 94 links. This comparison does not only reflect the consistency of the editors, but also provides a rough upper bound on how well the classifier could do.

california lottery(**)	table tennis(**)
aerospace defence(**)	weather(**)
IBM research center(**)	
web browser(*)	rental car(*)
jennifer lopez(*)	super bowl(*)
art history(*)	web proxy(*)
translation online(*)	trim spa(*)
picnic(*)	hand games(*)
US open tennis(*)	wine(*)
image processing(*)	teen health(*)
healthcare(*)	
online casino	IT company
source code download	humanities
native+tribal	theatre
kids entertainment	library
education reference	party games
ask an expert	gifts shopping
music shopping	pets shopping
business service	small business
E-commerce	wholesale
healthcare industry	chemicals
mental health	addictions
health insurance	dentistry
breaking news	weblog news
car buying	TV channel
tennis games	food drink
rebate online	stocks
chinese web portal	local search
mtv download	morning call
wall street	music channel

Table 1: Set of queries used for collecting query-specific data sets.

5.3 Link classification

Based on the set of links that are human labeled, a linear SVM classifier is trained and tested using SVM^{light} [8]. The 1016 labeled samples (undecidable links are excluded) are randomly split into two halves, on which a two-fold cross validation is performed. The average accuracy is 83.8%. The precision and recall of positive class (qualified links) are 71.7% and 82.2%, respectively. The trained model shows that anchor text similarity is the most discriminative feature, followed by non-anchor text similarity.

To find out how discriminative the anchor text similarity is, we trained and

tested a linear svm classifier on the anchor text similarity only. The average accuracy is 72.8%, significantly lower than that using all the six features.

We also trained a classifier on the whole labeled set and tested it on the same data so that we can get an optimistic estimation of the classifier's quality. The accuracy is 85.1%, with precision and recall being 73.7% and 83.4%.

In order to get a better insight into the features, we plot the human-assigned labels to feature values in six graphs (Figure 4 through Figure 9), each showing one of the features. For each feature, the range of the feature values is equally divided into 20 subranges (or, buckets). In each graph, x-axis depicts the set of value ranges. The bar graph shows the distribution of that feature of all human-labeled links. The line graph shows the percentage of qualified links in each range.

From Figure 4, we can see that the distribution of topic vector similarity is somewhat polarized, with the majority gathering at the first and last range. This is because the topic vector given by the textual classifier is polarized. In most vectors, one component dominates others. As a result, the cosine similarity of two vectors tend to be quite close to zero or one. The recurrent fluctuation of the probability of "qualified links" indicates that topic vector similarity is not a good feature for detecting "qualified links".

Compared with the distribution of topic vector similarity, the distributions of the three content based features (content similarity, anchor text similarity, and non-anchor text similarity, shown in Figure 5, Figure 7, and Figure 8, respectively) are more smooth. About the probability of "qualified links", although there are still minor fluctuations, the high probability within the first three or four buckets, followed by a dramatic decrease from the fifth to seventh bucket, shows that the links in the rear buckets are mostly "unqualified links". This result indicates that links between two pages that are too similar are likely to be "unqualified". This matches our observation in practice on navigational links and spam links, where the source and target pages usually have a large portion of content or anchor text in common.

The results on host name similarity and URL similarity, shown in Figure 6 and Figure 9, are not interesting. They are easy to compute (they can be computed without the content of the page), but their usefulness in "qualified link classification" is also limited.

5.4 Retrieval performance

The classification of links is only an intermediate step. The final goal of qualified link analysis is to improve retrieval performance. Here, we test Q-HITS on the query-specific datasets, and compare its result with that of Bharat and Henzinger's *imp* algorithm [1]. Since five of those datasets have been used for human labeling of links, the remaining 53 query-specific datasets are used for the evaluation of retrieval performance.

Since there is no available evaluation for results of these query-specific datasets, the relevance between query and search results have to be inspected manually. In our evaluation system, the top ten search results generated by

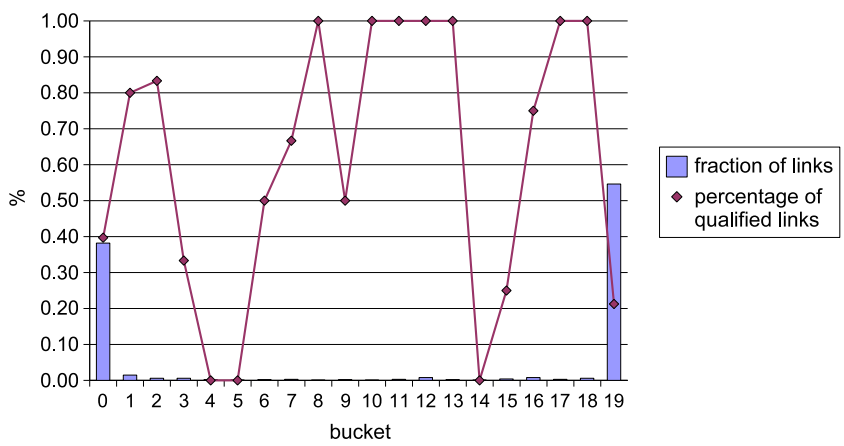

Figure 4: Topic vector similarity

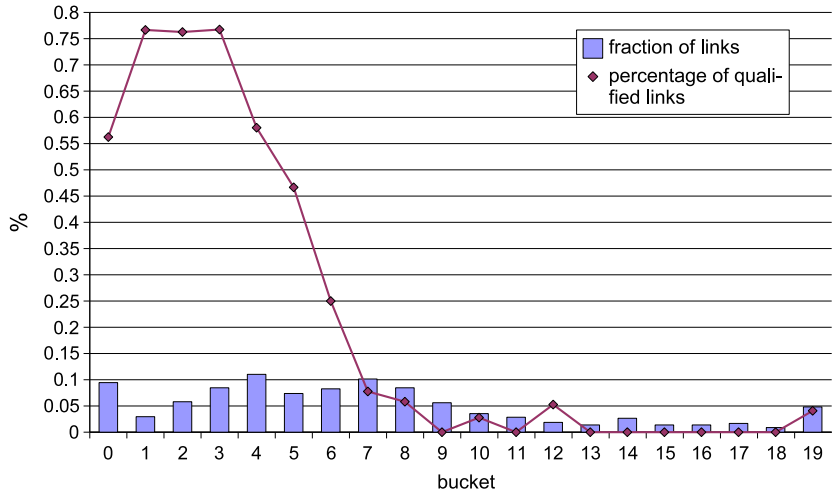

Figure 5: Content similarity

various ranking algorithms are mixed together. To evaluate the performance, 43 participants are enlisted, to whom a randomly chosen query and a randomly selected set of ten results (of those generated for the given query) were shown. The evaluators were asked to rate each result as quite relevant, relevant, not sure, not relevant, or totally irrelevant, which were internally assigned the scores of 2, 1, 0, -1, -2, respectively. A page is marked as relevant if its average score is greater than 0.5.

Based on the evaluation data, we can calculate the overall precision at 10 (P@10) for each approach; in addition, the overall average relevance score

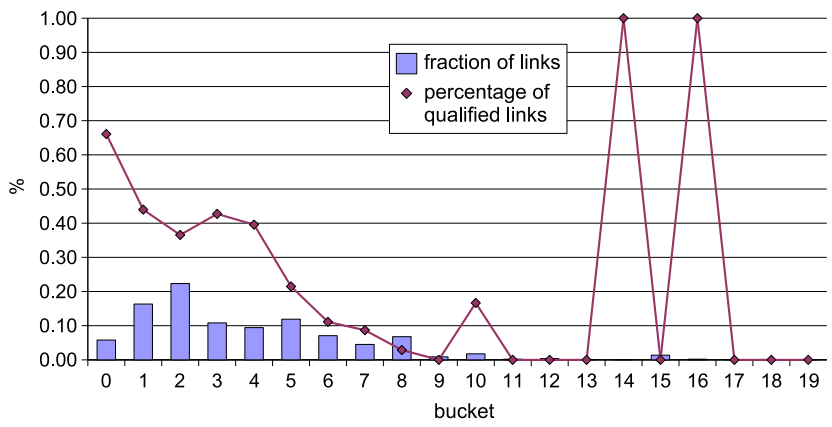

Figure 6: Host name similarity

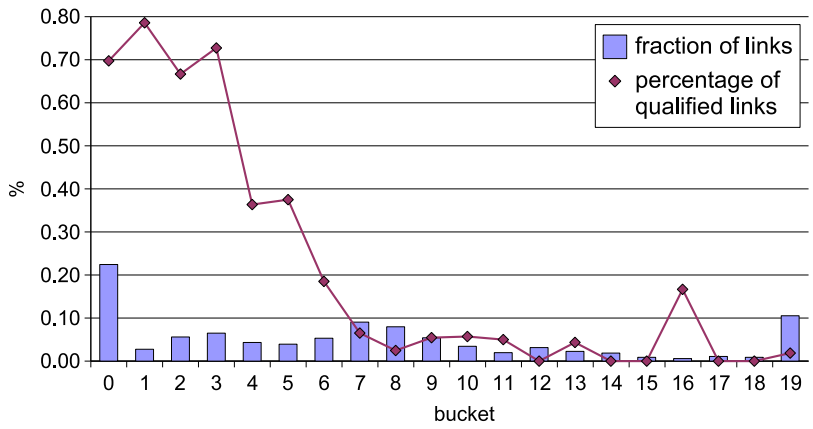

Figure 7: Anchor text similarity

(S@10) is calculated to further explore the quality of retrieval since precision cannot distinguish high-quality results from merely good ones. We used these two metrics to compare the performance of the different approaches.

5.4.1 Sample search results

Here we first demonstrate the striking results this technique makes possible by an example query "US open tennis". In Table 2, the top 10 results returned by *imp* are dominated by a group of touring and traveling pages that are strongly connected. After applying Q-HITS, the links inside this community is broken. The undeserved authority of its members are reduced.

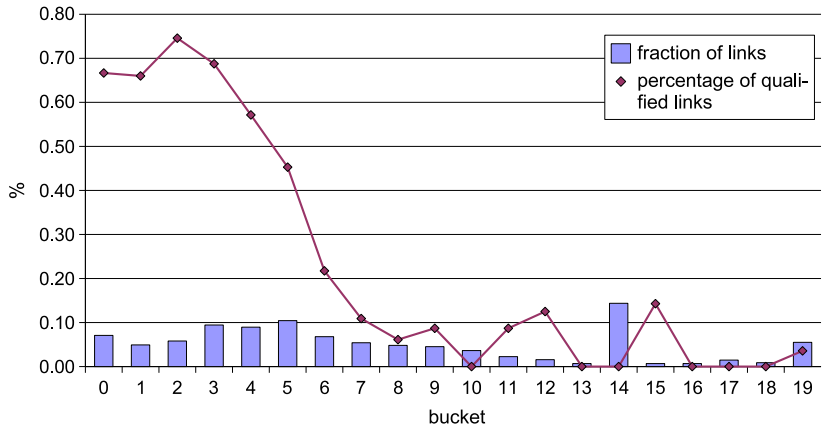

Figure 8: Non-Anchor text similarity

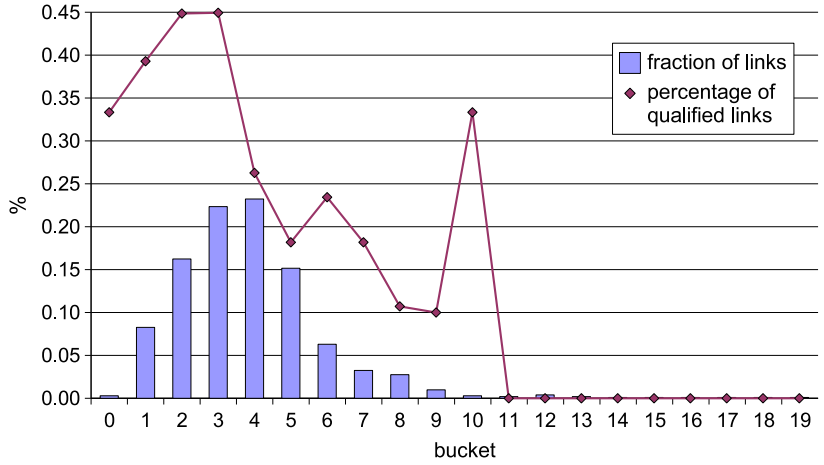

Figure 9: URL similarity

5.4.2 Evaluation

Figure 10 shows the comparison of original HITS, *imp*, and Q-HITS. The average precision of the top 10 results (precision@10) of HITS is only 0.38. *imp* improved that by a large difference to 0.69. By filtering out "unqualified links", precision@10 can be further improved to 0.75; the average score is improved by almost one third from 0.74 to 0.96, comparing with *imp*. T-tests show that the improvement of Q-HITS over *imp* in precision and score are both statistically significant (p-values are 0.024 and 0.012, respectively).

We also compared our approach with the spam detection work by Wu and

Rank	URL
1	http://www.luxurytour.com/
2	http://www.rivercruisetours.com/
3	http://www.escortedtouroperators.com/
4	http://www.atlastravelweb.com/
5	http://www.atlastravelnetwork.com/
6	http://www.sportstravelpackages.com/
7	http://www.atlasvacations.com/
8	http://www.escortedgrouptours.com/
9	http://www.escorteditalytours.com/
10	http://www.atlascruisevacations.com/

(a) Top 10 results by *imp*

Rank	URL
1	http://www.tennis.com/
2	http://www.usopen.org/
3	http://www.wtatour.com/
4	http://www.usta.com/
5	http://www.atptour.com/
6	http://www.itftennis.com/
7	http://www.frenchopen.org/
8	http://www.gotennis.com/
9	http://www.tennistours.com/
10	http://www.sportsline.com/u/tennis/

(b) Top 10 results by Q-HITS

Table 2: Results for query *US open tennis*.

Davison[13] (denoted as "Link farm removal") on the 15 queries in common (marked with * in Table 1). The result is shown in Figure 11. On those 15 query-specific datasets, the precision@10 of HITS is 0.30. Link farm removal boosts that to 0.65. Having a precision@10 of 0.78, Q-HITS outperforms "Link farm removal" by 20%.

In "Link farm removal" algorithm, the links among identified link farm members are dropped. We compared the links dropped by Q-HITS (i.e., unqualified links according to the classifier) and the links that are dropped by "Link farm removal". Figure 12 shows the result of this comparison. Q-HITS dropped 37.17% of all the links; "Link farm removal" dropped 18.93%. The intersection of the links drop by the two algorithms accounts for 17.30% of all the links.

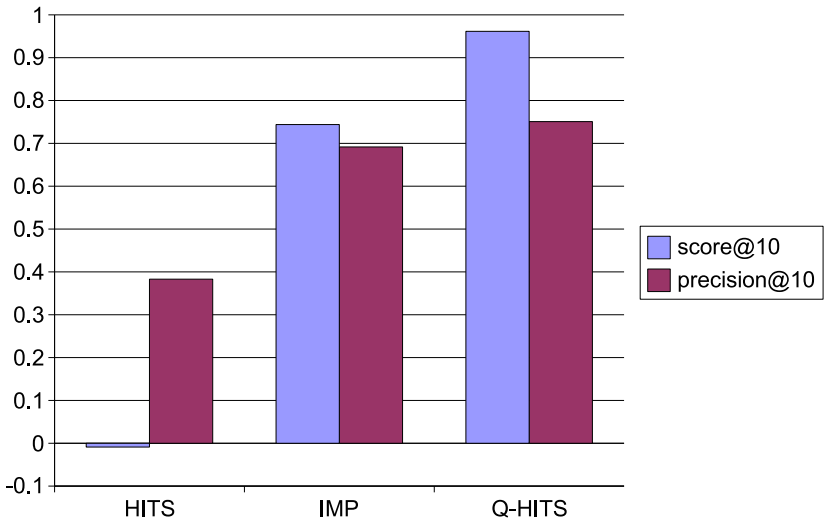

Figure 10: Retrieval performance on 53 query-specific datasets

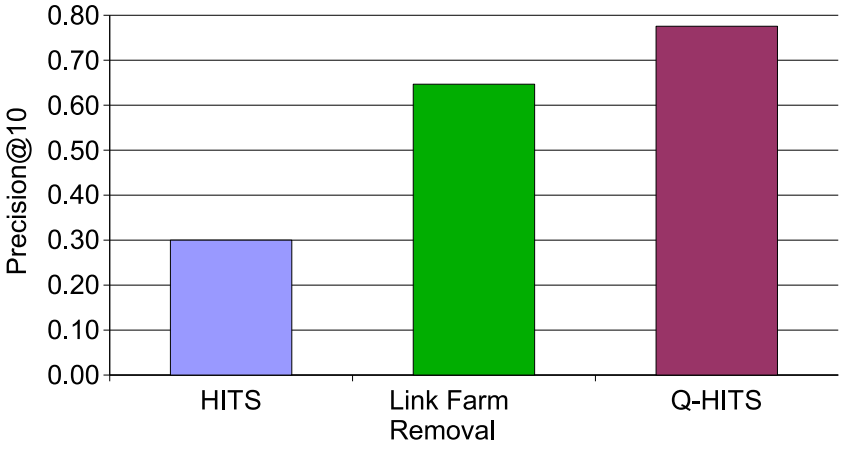

Figure 11: Retrieval performance on 15 query specific datasets

6 Discussion and conclusion

In this paper, we presented the approach of identifying qualified links by computing a number of similarity measures of their source and target pages. Through experiments on 53 query-specific datasets, we showed that our approach improved precision by 9% compared to the Bharat and Henzinger *imp* variation of HITS.

This paper is merely a preliminary study, demonstrating the potential of our

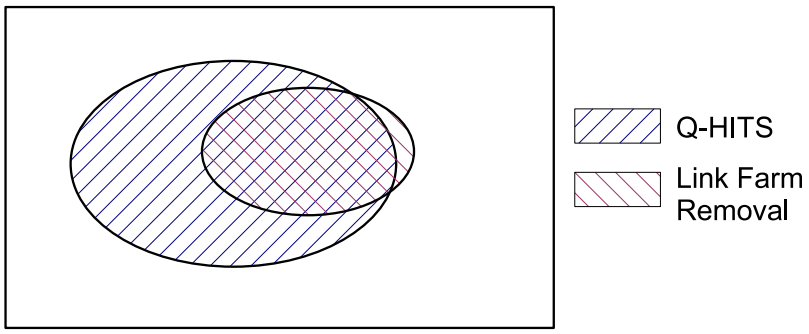

Figure 12: Fraction of dropped links

approach. The following limitations can be addressed in future work.

- The classifier and similarity measures being used are quite simple. It is expected that the use of a better classification algorithm and an advanced set of similarity measures would produce a better result.
- The punishment of removing "unqualified links" might be too stringent. The experimental results show that there are some authoritative pages being removed. Weighting the links by their quality could be a better alternative than pruning.
- The computational complexity of "qualified link analysis" is an issue that needs to be carefully considered. Although the index of the corpus could be made available before hand, computing the similarity scores is still not an easy job considering the size of the web. Potential solutions include using fewer features, using features that are easy to compute, and utilizing simple classification algorithms. One possible extension that we have tested is to build a thresholding classifier based on the anchor text similarity. The classifier simply categorizes the links within the first eight buckets as "qualified links", and the rest as "unqualified". This approach gives a precision of negative class ("unqualified links") at 97.05% on the labeled training set. This classifier is then applied to the 53 query-specific datasets. The retrieval performance is between that of *imp* and Q-HITS (precision@10 being 73.02%, score@10 being 0.93).
- We only showed improvements of qualified link analysis on query-specific datasets. A test of our algorithm on global datasets is still needed.
- This approach is not a panacea for "unqualified links". We did not differentiate the different types of "unqualified links". Some types are perhaps more difficult to identify than the others. Perhaps some should have been less severely punished than others. A fine-grained discrimination could further boost the retrieval quality.

Acknowledgments

We thank Baoning Wu for helpful discussions and providing the datasets. This work was supported in part by a grant from Microsoft Live Labs ("Accelerating Search") and the National Science Foundation under award IIS-0328825.

References

[1] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, Aug. 1998.

[2] 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, Brisbane, Australia, Apr. 1998.

[3] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proceedings of the 10th International World Wide Web Conference*, pages 211–220, 2001.

[4] B. D. Davison. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, pages 23–28. AAAI Press, July 2000. Presented at the AAAI-2000 workshop on Artificial Intelligence for Web Search, Technical Report WS-00-01.

[5] I. Drost and T. Scheffer. Thwarting the nigritude ultramarine: learning to identify link spam. In *Proceeding of the ECML*, 2005.

[6] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of WebDB*, pages 1–6, June 2004.

[7] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, 2004.

[8] T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

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

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

[11] L. Li, Y. Shang, and W. Zhang. Improvement of HITS-based algorithms on web documents. In *The eleventh International World Wide Web Conference*, pages 527–535. ACM Press, 2002.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Unpublished draft, 1998.

[13] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference*, pages 820–829, Chiba, Japan, May 2005.

[14] Yahoo!, Inc. Yahoo! `http://www.yahoo.com/`, 2006.

[15] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy. Making eigenvector-based reputation systems robust to collusions. In *Proceedings of the Third Workshop on Algorithms and Models for the Web Graph*, Oct. 2004.