Hierarchy in Web Page Similarity Link Analysis

Allan M. Schiffman
Carnegie Mellon University &
CommerceNet Labs

CommerceNet Labs Technical Report 06-02
May 2006

Abstract

Rather than using traditional text analysis to discover Web pages similar to a given page, we investigate applying *link analysis*. Since web pages exist in a link-rich environment, that has the potential to relate pages by any property imaginable — since links are not restricted to intrinsic properties of the page text or metadata.

In particular, while Web page similarity link analysis has been explored, prior work has deliberately ignored the explicitly hierarchical host & pathname structure within URLs. To exploit this property, we generalize Kleinberg's well-known "hubs and authorities" HITS algorithm; adapt this algorithm to accommodate hierarchical link structure; test some sample web queries; and argue that the results are potentially superior and that the algorithm itself is better motivated.

Acknowledgments

This report draws upon the author's research for *Dynamic Network Analysis* with Prof. Kathleen M. Carley in Spring 2006. See http://www.cs.cmu.edu/~jhm/PEDescriptions/Carley_Intro_DNA.htm

Additional editorial input from Rohit Khare.

1. Introduction

When we find something we like, we want more of it. So, given a document that proves enjoyable or useful, one may ask, "Where can I get more documents like this one?" Such a query, for *related* or *similar* data, has been considered for database systems for thirty years [3] and continues to be an area of active research [4].

We are interested in document similarity in the context of the Web [5]. Our original motivation was due to the CommerceNet *Wowbar* project [7]. Wowbar is a web-browser sidebar that presents material a user might be interested in that is associated with the main web page the user is currently viewing. A summary of pages[1] similar to the main one would be a particularly useful data source for Wowbar.

1.1 Approaches to Similarity

Search engines have traditionally supported similarity queries and do so today, but beyond a description of Alexa.com's facilities circa 2000 [24] we have not found a useful description of current implementations.[2]

Stated abstractly, responding to a similarity query involves:

1. generalizing from a specific document by extracting features in or related to it
2. analyzing a corpus of documents for matches against those features
3. ranking candidate documents with respect to those features from the corpus
4. presenting a few of the documents ranked as 'most similar'

Features to be used as part of a similarity query either can be *intrinsic* to the document itself, or *extrinsic:* how the document relates to the rest of the world.

1.1.1 Intrinsic: Text and Metadata Analysis

Intrinsic document features are derived from the data in the document itself. For a text document, we could use a word or phrase list; for an image, some characterization of the image's colors.[3] Such techniques are commonplace and are an area of active research, see, for example, the literature on *Latent Semantic Analysis* [9].

---

[1] In this paper we refer to the original data used as the basis for the query, or the results of the query, as a *document*, and in the context of the Web, as a *page*, although it could conceivably be a specialized data structure or encoded data representing a rich media type such an image or audio/video recording.

[2] The Netscape browser's "what's related" menu originally relied on Alexa's similarity results.

[3] Ranking image similarity by overall color might not seem useful, but there are image retrieval systems which do exactly that – e.g., FotoFile [8].

Another source of features for similarity analysis is document *metadata*: information stored by the data repository about the document.[4] Metadata that might be useful in a similarity query include the author or the creation date of the document. These features are used in queries so common we might not think to classify them as similarity queries: "show me books by the same author", or "show me yesterday's stories".

1.1.2 Extrinsic: Web Link Analysis

Link- or network-analysis algorithms have the prospect of being scalable, language and media -independent, as well as robust in the face of link-spam [15] and topic complexity. The potential superiority of link analysis is easily illustrated by some examples. How, other than by link analysis might we expect to find out that:

- www.gm.com is the home site for a company that might go bankrupt?
- www.etniesbmx.com/site-images/news/article/pic00153.jpg is a scary picture?
- www.globalaigs.org is a very badly designed webpage?

In each of these cases no text on the page would suggest such a classification (indeed, the image has no text at all), yet there could be (and are) many other pages that link to the above-named pages that imply just such classifications, explicitly suggested or not. A link to a page is not just a generic expression of interest, but also an indication as to how the page should be grouped with others.

One need not exclusively rely on either text or link analysis for similarity queries; hybrid strategies are possible and have been proposed [16]. We believe link analysis has more potential for real novelty — to get more out of the document than a page creator's original words by harnessing the associations within other creator's minds.

1.2 Project Overview

Our work generalizes Kleinberg's well-known "hubs and authorities" HITS (Hypertext-Induced Topic Selection) algorithm [11], adapts the algorithm to accommodate link structure, tests a few queries against real web data, and interprets the results.

The original plan was to create a variant implementation of Kleinberg's HITS that altered the graph analysis by introducing *phantom nodes* corresponding to URL components for nodes otherwise in the graph. Upon reflection, it seemed unnecessary to test this phantom node concept simply to contrast with HITS and its variants, since prior work eliminated any use of URL pathname hierarchy information whatsoever.

---

[4] Actually, while some metadata could be said to be intrinsic and derivable from the document itself (such as its length), some other metadata, as defined here in the sense of being notionally contained by the repository, could be extrinsic – such as retrieval frequency (popularity). We gloss over this distinction as unimportant here.

1.3 Contributions

This project makes several contributions:

- We create a generalization of the HITS equations that permits for accounting of node and link properties. Similar generalizations could be made to related algorithms such as SALSA [12].
- We design a (hub) node weighting function that accounts well for node out-degree.
- We design a specific link weighting function that accounts for hierarchical containment relationships determinable by link (URL) structure.
- Contrary to other published versions of HITS and its variations, given the link weighting accommodation described in (3) above, we are able to retain inter-page link relationships that occur within a single domain.
- We adapt the HITS subgraph-construction step from keyword queries to similarity queries; the Kleinberg paper only hints at this.

Of these, we consider the retention of link relationships within a single DNS domain to be the most significant development in this project.

2. Algorithms Investigated

We describe here the algorithms used in the remainder of the paper. Our notational conventions are as follows. We define a directed graph *G* as made of the set of vertices V (of order *g*) and the set of edges E the elements of which are pairs of vertices defining a directed edge:

$$(1) \qquad G = \langle V, E \rangle; g = |V| \quad e \in E, u \in V, v \in V : e = (u,v),\ (v,u) \neq (u,v) : u \neq v$$

We define for a vertex *v*, the set of its outgoing (forward) links as F_v, and its incoming (backwards) links as B_v:

$$(2) \qquad F_v = \forall u : (v,u) \in E \quad \text{and} \quad B_v = \forall u : (u,v) \in E$$

2.1 Kleinberg's HITS: overview and modifications

Kleinberg's HITS algorithm models pages on the web as a collection of *hubs*, which are pages that link to interesting or useful pages; and *authorities*, which are interesting pages that are linked-to by hubs. This circular definition leads to an iterative algorithm that ranks all nodes in a (sub)graph according to their qualities as a hub and as an authority.

An interesting property of the web that this model accounts for is competition. A directory page (hub) on Yahoo or DMOZ might not link to their competing directory, but would link to an authority page such as Sony or Apple. In turn, those authorities will likely not link to each other, and probably will not link to most of the pages that reference them.

2.1.1 Constructing the Base Set subgraph

Before HITS can run its iterative algorithm to rank nodes (described below), it must select nodes to rank. Kleinberg refers to this phase as *sampling* and building the *base set subgraph*. In the original paper, which was oriented to keyword queries rather than similarity queries, this was accomplished by a two-step process. The first step runs the keyword query proper, which the paper calls building the root set.[5] This small set of pages is expanded to the base set to be analyzed by recursively fetching the forward and backward links from the accumulating subgraph up to some cutoff distance from the root set. Then this set of collected nodes is ranked and thresholded (to the first ten or so) best hub and authorities.

2.1.2 The HITS recurrence equations

The original HITS equations are a pair of recurrence relations, where for each node in the directed graph, a hub-score *h* is computed as the sum of the authority score *a* of each of its neighbors (children); in turn, the authority score is computed as the sum of the hub scores of its neighbors (parents):

$$a_p = \sum_{q:(q,p)\in E} h_q \quad \text{and} \quad h_p = \sum_{q:(p,q)\in E} a_q \tag{3}$$

…these recurrences are iterated in an alternating fashion until the scores converge, often in less than a dozen steps.[6]

2.2 A Generalization of the HITS Equation

We have recast these recurrence equations into a generalized form, parameterizing variants to HITS such as suggested in Chakarabarti et.al. [1] and elsewhere [2]. In our generalized form, we have a link-weighting parameter ω and node-weighting parameters κ and α for hubs and authorities, respectively:

---

[5] The original paper (written before PageRank or Google) assumed the existence of text search services such as Alta Vista which used text analysis (relevance) rather than link analysis (support) ranking.

[6] Actually, steps to convergence is related to graph diameter, but typical use of HITS deals with graphs in a modest range of diameters. Kleinberg's original paper [11] provided a proof of convergence by noting the algorithm computes the principal eigenvector of the adjacency matrix.

(4) $$a_p = \kappa_p \sum_{q,e:e=(q,p)\in E} h_q \omega_e \quad \text{and} \quad h_p = \alpha_p \sum_{q,e:e=(p,q)\in E} a_q \omega_e$$

2.2.1 Node Weight Functions

The literature suggests little in the way of useful authority-node weight functions (α), and no compelling ones have occurred to us (yet).

On the other hand, hub-node weighting (κ) that penalizes out-degree (as suggested by many proposed variants of HITS) has a compelling purpose. A hub should not get a vastly improved score simply by having a large number of links. Intuitively, one might value a page with a small number of good references more than a page with a large number of bad references. One could simply weight the hub score as the inverse of out-degree:

(5) $$\frac{1}{|F_p|} : (p,q) \in E$$

This has been proposed, but would not capture the notion that quality of links being equal, more links are probably better than less. So, we choose a κ that sub-linearly penalizes out-degree:

(6) $$\kappa_p = \frac{1+\log(|F_p|)}{|F_p|} : (p,q) \in E$$

2.2.2 Link Weighting

A potentially important link weighting measure, and central to this paper, is to properly penalize *path containment*, which we define here. The original Kleinberg paper, and almost all its successors (see Borodin et.al. [2]) dismiss the possibility that intra-domain links[7] can do anything but degrade the results of ranking algorithms and dispense with them completely.[8]

Justification for dispensing with links that share naming components between parent and child page has been justified for several reasons starting from the notion that ranking based upon link analysis considers a link from one page to another as a sort of "vote of support":

---

[7] Domain here refers to the Domain Name System [6], that is, Internet host names such as www.cmu.edu or gmail.com. An intra-domain link means a page like http://www.cmu.edu linked to http://www.cmu.edu/home/pros_students/index.html

[8] For instance, the original Kleinberg paper [11] refers to intra-domain links as intrinsic ones and deletes them from the graph entirely before processing (page 6 of the paper).

1. *Artificial Support*: Links from one part of a site to another possibly indicate an effort to artificially boost link ranking (see Spamdexing [15]), or at the very least may be expected to be biased in that all the pages of the site "belong" to a single authority.
2. *Navigation, not Support:* Intra-site links that are intended to simplify site navigation cannot be distinguished from other links. All the pages in an online dictionary may have a link to "home", but that does not mean that the title page is, say, 30,000 times more interesting than a page for any individual word.

We call these two (partially misplaced) concerns the *nepotism* problem. Disregarding link information simply because the links share common components (or just a common domain) is too simplistic; it does not solve nepotism and eliminates important link information.

2.2.2.1 Link Path Hierarchy and Containment

We have noted that URLs have structure, inspired by pathnames of disk file systems. From the point of view of the party making the reference (and by the link analyst), we might say URLs are without structure or semantics since the referrer likely does so without considering the URL proper.[9] In computer science parlance. the reference is opaque.

But URLs do have structure, and that structure gives clues about the page itself, in the text analysis sense, but also in terms of what can be inferred about relationships to other pages. We define hierarchical *path containment* as the condition where one URL's path components are a leftmost prefix of the other's. So we say that cmu.edu/a/b is contained by cmu.edu/a.

We wish to account for the relationship between nodes that share pathname components, and note that by not discarding references between pages in the same domain we can capture such information (without requiring the phantom node concept we originally contemplated). Since web pages often have embedded links that follow the path hierarchy (parent pages pointing to children and/or children pages pointing to parents), explicitly available graphs will capture the containment relationship between pages and our link weighting mechanism can account for that relation.

2.2.2.2 Interpretations of Containment

It is an error to assume that if a path is contained by another than they both share common agency, authorship, trust boundaries, administrative boundaries or the like. And the opposite is no more true.[10]

---

[9] Without structure or semantics, that is, beyond that of that specified by the URL syntax standard: i.e., scheme, netloc, path, query, fragment.

[10] For example, www.gmail.com is administered by the same parties that administer www.google.com; in fact it is an alias for mail.google.com. So having different domain names doesn't assure the lack of "nepotistic" ties.

By way of illustration, webpages.charter.net/allanms and webpages.charter.net/dcarlsen are different user's home pages, with nothing in common other than ISP. And these are not unusual examples, but commonplace ones considering the largest web presences (measured by either hits or number of users) include sites such as Blogger, Myspace and Facebook, which use exactly this sort of URL structure.

A link from one page in myspace.com to another page is not a nepotistic self-reference by some Myspace employee attempting to boost rankings for the site, but rather a citation of one author's page of another's.

Since we consider link containment to only weakly indicate that the pages are related in some "nepotistic" way, we use a weighting function to account for a range of possibilities based upon containment depth.

2.2.2.3 Link Containment Weighting Function

We can state the boundary and recurrence conditions for our link-weighting, given a damping factor *d*: 1) links that are equal have weight of zero; 2) links that have no components in common have a weight of unity; 3) links that have one leftmost component in common have a weight of *d*; 4) links with *n* leftmost components in common have a weight of d^n.

We present the function to compute link weight here below in an imaginary typed dialect of Python. Function `link_weight` returns the weight by comparing the component strings of the (url) vertices. The arguments are two arrays of strings, which are the URLs of the edge nodes separated at the component delimiters defined by URL syntax [14].[11]

```
def link_weight(array of string uc1, uc2):
     if uc1 == uc2: return 0.0
     w = 1.0
     for i from 0 to min(len(uc1), len(uc2))
          if uc1[i] equals uc2[i]: w *= 0.5
          else: break
     end for
     return w
end def
```

We set the damping factor here (and in our experiments) arbitrarily at 0.5; we suspect this under-weights links but have not attempted any sensitivity analysis.

---

[11] The reader is forgiven for assuming that the URL component delimiter is the forward slash (/), but the situation is more complex than that. Urls may contain *queries* that are delimited by '?', *fragments* delimited by '#', and the *scheme* that is delimited by '//:'. Domain names likewise have pathname-like structure (read from right-to-left and delimited by dots); one could consider these as components too, but we have not done so here.

2.2.3 Other modifications made to HITS

Although this paper's primary contribution is the hub and link weighting schemes given above, we have made some other minor adaptations of HITS to suit the circumstances.

2.2.3.1 Constructing the base set

In our application of HITS, we are performing a similarity search rather than a topic search, so our base set is simply the subgraph surrounding the target page at some fixed radius. As described in section 3 below, our experiments used an internal graph database representation of the web in our regions of interest. So, constructing the query base set means an efficient set of queries to a local database keyed by URL. This is typically completed in less than a minute, as opposed to than crawling the actual web, which could have conceivably taken hours per query.

2.2.3.2 Suppressing duplicate nodes in ranking output

As a way to compensate for pages in the results that are very similar, when delivering the list of ranked nodes we optionally suppress adjacent entries from the same domain. This is less necessary for results of weighted HITS.

2.3 Similarity Ranking by Hamming Distance

In order to have a more traditional similarity metric to compare our modified HITS results with, we choose a direct measure of structural equivalence and rank nodes in the base set subgraph by Hamming distance (see Banks and Carley [13]) from the target node. Although we have a graph representation of the base set rather than an adjacency matrix representation, we can compute Hamming distance *H* directly (using Δ to mean set symmetric difference):

$$H(u,v) = \left|F_u \Delta F_v\right| + \left|B_u \Delta B_v\right| \tag{7}$$

That is to say, the Hamming distance between two vertices is the order of the set symmetric difference of the neighbors for the two vertices.

3. Methodology

We tested several variations of HITS against a small number of queries at several base set radii. We describe here the steps required to allow these investigations to leave behind slow, bandwidth-limited, error-prone, and irreproducible web queries in favor of fast, reliable and repeatable graph algorithms.

3.1 Crawling the Web with Itsy

This project required the results of a web crawl, and rather than choosing an existing crawler, we developed our own, which we call *itsy*. Existing crawlers invariably save the entire resource fetched to the crawler's local disk.[12] For link analysis, we don't have any use for the page data itself once links have been extracted. It is also more convenient to analyze explicitly represented link data than to repeatedly analyze web pages, even if they are cached locally.

Itsy's job is crawl the web by fetching pages. After extracting links, it discards the fetched page and stores the results as a explicit representation of a page node in a custom web graph database. Each node in the database has explicit fields for inbound links, outbound links and page statistics (such as time of last fetch).

An important itsy capability is to creating *synthetic crawls*. Since we don't actually need page data, only page features, itsy can make special queries and store the analysis. Specifically, itsy uses Google's `link:` directive to determine the inbound links to a given page [23]. This allows itsy to gather data it would be impractical to obtain accurately otherwise, since determination of all inbound links to a page potentially require scanning the entire web. The result of an itsy spiral crawl allows a link analysis program to make high speed, efficient queries on a local node database.[13] We note that itsy performs the same function as the *Connectivity Server* [21], a system that was used to develop and test the *Companion* algorithm [22], a derivative of HITS that was explicitly similarity-oriented.

3.1.1 Practical details and special cases

Itsy is about 1400 lines of Python code. It obeys the `robots.txt` crawler exclusion conventions and rate-limits to one fetch per second to avoid overwhelming sites with fetch requests.

Not all web pages are considered equal by itsy; special rules are applied both during crawls and in a post-processing step.

3.1.1.1 Rules applied during crawls

Since we are interested in building a pagenode graph, we don't want to fetch any pages that don't contain links. Therefore, itsy only makes requests for HTML files, and then only by the HTTP access method. Other file types are discarded, even if they could conceivably be scanned for anchors, for ease of implementation.

---

[12] There are many open source crawlers to choose from, such as Nutch, Harvest and Sphinx [20].

[13] *Spiraling* is our term for the standard itsy operation where, given a set of URLs, it does a breadth-first crawl on inbound and outbound links, out to a specified radius from those URLs.

Itsy maintains a table to exclude domains by name, if requested by appropriate authorities. This is in addition to the usual rules for spider exclusion and host rate limiting. No site administrator requested to be excluded from our crawls during our tests. Another table is used to exclude domains by name that we have determined to be irrelevant or problematic; twenty domains were eventually put on this list.[14]

3.1.1.2 Post-processing rules

A separate database scrub pass is periodically performed to remove any inconsistent nodes or malformed URLs that are in the database because of bugs in itsy, crashes during crawls, and malformed links in fetched pages.

3.2 Constructing a Base Set with Itsy

When we have a large enough crawl to deal with the queries we plan to make, itsy itself no longer plays a role and we access the page-node database where itsy stored its results. We use a separate set of classes in this phase.

Graph and *Node* are entirely in-memory structures that correspond to itsy's database-oriented classes *PNdb* and *PageNode*. Where *PageNodes* represent links as URL strings, *Nodes* have references to other Nodes; and where *PNdb* uses URLs as a key in a disk-based Btree, *Graph* uses a hash table. Constructing a HITS base set thus requires walking the *PageNode* database, building *Nodes* along the way, and linking them together to return the *Graph*.

3.2.1 Choosing a base set radius

Recall that the first step of performing a hubs-and-authorities node ranking is to construct a base set (a sub-graph relevant to the query), and for a similarity query that does not perform text analysis, we simply take the subgraph in the vicinity of the target page. The only free parameter, assuming forward and backwards links are treated symmetrically, is: what radius from the target node do we select graph nodes for the subgraph.

The minimum non-singular radius of 1 could conceivably be interesting, but would not be likely to select many candidate nodes. Surprisingly, large radii are also not useful for similarity ranking, and not simply because they would increase algorithm running times.

Large radii would result in generalization, or drift away from relevance to the target page and towards the foci of topics for the graph as a whole. This is easy to see when one considers that as radius increases to approach the diameter of the graph, the constructed subgraph of

---

[14] Advertising service domain names such as doubleclick.com and overture.com are on the list, as well as names clearly in error such as the 127.0.0.0 IP loopback address.

any two target nodes has more nodes in common. And of course, subgraphs that tend to be equivalent tend to have equivalent node rankings. Therefore, at even modest radii in well-connected graphs, HITS on wildly different target pages would yield identical top-rated results!

In the literature, base sets are typically constrained to contain only a few thousand nodes. For a typical similarity target subgraph, this translates to radii in the range two to five.

4. Results

We describe here the results of some test queries we performed in April 2006. No attempt was made to quantify (or improve) the performance of any step in our methodology, but a few observations are worth making here:

- Crawling averages approximately one second each page in our configuration, not counting manual error recovery.
- Conversion from disk database to in-memory internal representation takes about a minute for medium graphs (~5K nodes).
- Running gHITS on medium graphs takes a few seconds.

We have not carefully analyzed the running time of any of these algorithms, but they seem linear in number of nodes.

4.1 Crawl Statistics

All of the work presented here was done using an itsy node database resulting from a week of supervised "spiral" crawls performed in mid-April 2005. The crawl was stopped and restarted many times, using a checkpointing mechanism that attempted to pick up where the crawl stopped. The initial crawls were on a subset of the pages mentioned in the Kleinberg paper,[15] with some additions of interest to the author such as:

webpages.charter.net/allanms	*Author's blog*
www.dmoz.org	*The Open Directory*
www.cmu.edu	*Author's university*
www.commerce.net	*Author's employer*
www.casos.cs.cmu.edu	*Affiliated w/ Author*

This result, which we call the "big crawl", had statistics presented in Table 1 below.

---

[15] Crawl starting points chosen to allow for comparison with Kleinberg's results included: www.eff.org, www.ford.com, www.toyota.com, www.audi.com, www.nyse.com, www.plannedparenthood.org, and www.caral.org among others.

Number of Nodes (pages)	91,103
Number of Node Domains (hosts)	27,111
Average Out-degree	51.7
Average In-degree	4.5
Pendants	26,664
Database size	624 MB

Table 1: Itsy's big crawl

The ten most frequently occurring domains were: dmoz.org, www.political.com, www.cdt.org, www.cnn.com, www.montclar.edu, www.survivalarts.com, www.stickyminds.com, www.casos.cs.cmu.edu, lists.w3.org, and www.wiesenthal.com.

4.1.1 Sample Queries

We performed two similarity queries for these URLs:

- http://webpages.charter.net/allanms — the *blog* query
- http://www.cmu.edu/ — the *school* query

In each case we used base sets constructed at radii of two, three and four from the target node. For each query, at each base-set radius, we ran gHITS with all four combinations of link and hub weightings, and presented the top twenty resulting hubs and authorities. In addition, for each of these radii, we presented the top twenty nodes ranked by minimum Hamming distance. A sample of results is presented in the Appendix.

4.1.2 Interpretation & Analysis

Unsurprisingly, the results for larger base sets (radius of three and four) seem clearly less relevant to the original query than smaller base sets. For example, the results of the blog query at radius four is quite similar to the school query!

We restrict our analysis here to the blog query at radius 2 (1926 nodes), which is presented in the appendix, and the school query at depth 3 (1153 nodes). We note the differences in results from the fully-weighted (hub and link) version versus classical, unweighted HITS.[16]

[16] Viewing the results in the appendix and in the attached files: the fully weighted result is the first part (of four) in a run, the unweighted is the last.

4.1.2.1 Weighted versus Unweighted HITS

The query results are subjectively plausible, in that they all meet the following test: if you were interested in the target page, you'll arguably be interested in the ranked result pages. More significantly, the weighted version of the algorithm returns superior results, objectively, with respect to the novelty of ranked results.

In the unfiltered output of the blogs query all twenty of the top-ranked hubs are from the same area of the same website and half of the top-ranked twenty authorities are from the same areas as the hubs, and the remaining ten are restricted to three sites.[17] The weighted version of the algorithm yields results that do not require filtering, exhibiting high relevance and diversity.

For the schools query, the difference between the weighted and unweighted algorithm is not as dramatic; however, the weighed algorithm still shows an improvement in relevance and diversity relative to the unweighted version.

4.1.2.2 Hamming distance results

We used nodal ranking by Hamming distance from target as a baseline measure, and the results are subjectively unsatisfying. In both the school and blog query, the Hamming distance ranking list nodes are disjoint from the hubs/authorities ranking list for all tested variants of the algorithm.

More subjectively, we might concede that the Hamming distance results are relevant enough, except on closer examination we note that the structural equivalence being detected (and used as a proxy for similarity) is based on commonalities that seem machine-centric and uninteresting. We note, for example, that the school query is close to other nodes that share similar "types" of links. If you use particular RSS feed and photo -sharing services and I use the same feed and photo -sharing services, that still doesn't imply we have much in common.

5. Conclusions and Future Work

The classical algorithms and the variations we tested provide plausible-enough results. The literature does not suggest any objective measures for similarity to test our algorithms against, so for now one must be satisfied with the author's and the reader's opinion of their usefulness relative to one another and in general. We could easily imagine experimenting with more variations and performing many sensitivity analyses on the algorithms, especially given that their running time is very modest.

---

[17] The unfiltered (not "picked") output was left out of the appendix in the interest of space, however, the same effects are quite noticeable in the filtered output, only not so dramatically.

There are several theoretical objections to hubs-and-authorities -style similarity search algorithms. It is particularly unsatisfying that increasing the size of the candidate corpus (the base set), which is the best way to increase the prospects for a novel query result, leads to unconstrained "drift" from relevance with respect to the target page. Adding relevance weighting to the recurrence equations (without resorting to text analysis) is quite feasible. Penalizing hubs/authority nodes that are only weakly connected (e.g., distant) to the target page would also be a good direction for future research.

References

1. Chakrabarti, Dom, Kumar, Raghavan, Rajagopalan, Tomkins, Gibson, and Kleinberg, *Mining the link structure of the World Wide Web*. IEEE Computer, **32**(8), August 1999.
2. A. Borodin, G. O. Roberts, J. S. Rosenthal, P. Tsaparas, *Finding Authorities and Hubs from Link Structures on the World Wide Web*, in 10th International World Wide Web Conference, May 2000.
3. Zloof, M.M., *Query-by-example: A data base language*, IBM Systems Journal, 1977, **16**(4): p.324
4. Maguitman, A. G., Menczer, F., Roinestad, H., and Vespignani, A., *Algorithmic detection of semantic similarity*. In Proceedings of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM Press, New York, NY, pp.107-116
5. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., and Secret, A. 1994. *The World-Wide Web*. Commun. ACM **37**(8), 76-82.
6. Mockapetris, P. and Dunlap, K. J., *Development of the domain name system*. 1988, In Symposium Proceedings on Communications Architectures and Protocols, Stanford, CA
7. Schiffman, A.M. *WowBar Notes*. 2005 Available from: http://wiki.commerce.net/wiki/Wowbar_notes.
8. A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka, FotoFile: a consumer multimedia organization and retrieval system. (1999) In Proc. ACM CHI Conf., pp 496–503
9. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, Indexing by Latent Semantic Analysis, Journal of the Society for Information Science, (1990) 41(6), 391-407

10. Haveliwala, T.H., et al. Evaluating Strategies for Similarity Search on the Web. in Eleventh Int'l World Wide Web Conference. 2002. Honolulu, Hawaii: ACM Press.

11. Kleinberg, J.M., Authoritative sources in a hyperlinked environment. J. ACM, 1999. **46**(5): pp.604-632

12. Lempel, R. and Moran, S. 2001. *SALSA: the stochastic approach for link-structure analysis*. ACM Trans. Inf. Syst. 19, 2 (Apr. 2001), 131-160

13. David Banks, Kathleen Carley, *Metric inference for social networks*, Journal of Classification, Volume **11**(1), Mar 1994, pp121-149

14. Berners-Lee, T., Fielding, R., and L. Masinter, *Uniform Resource Identifiers (URI): Generic Syntax*, IETF RFC 2396, August 1998.

15. Wikipedia, c. *Spamdexing*. [Wiki] 2006 Available from: http://en.wikipedia.org/w/index.php?title=Spamdexing&oldid=41459864.

16. Chirita, P.-A., D. Olmedilla, and W. Nejdl, Finding Related Pages Using the Link Structure of the WWW, in Proceedings of the Web Intelligence, IEEE/WIC/ACM International Conference on (WI'04) - Volume 00. 2004, IEEE Computer Society.

17. Fogaras, D. and R. Balazs, Scaling link-based similarity search, in Proceedings of the 14th Int'l conference on World Wide Web. 2005, ACM Press: Chiba, Japan.

18. Wangzhong, L., et al., Node similarity in networked information spaces, in Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research. 2001, IBM Press: Toronto, Ontario, Canada.

19. Hanneman, R.A. and M. Riddle. *Measures of similarity and structural equivalence*. Introduction to Social Network Measures 2005 Available from: http://faculty.ucr.edu/~hanneman/nettext/C13_%20Structural_Equivalence.html.

20. Cho, J. and Garcia-Molina, H. 2002. *Parallel crawlers*. In Proceedings of the 11th international Conference on World Wide Web, Honolulu, Hawaii, ACM Press

21. K. Bharat, A. Broder, M. R. Henzinger, P. Kumar, and S. Venkatasubramanian. *The Connectivity server: Fast access to Linkage Information on the Web*, (1998) In Proceedings of the 7th International World Wide Web Conference, pp 469-477

22. Dean, J., and Henzinger, M. R., *Finding related pages in the World Wide Web*, (1999) In Proceedings of the Eighth International World Wide Web Conference.

23. Google, *Google Web Search API*, 2006, available from: http://www.google.com/apis/reference.html

24. Lieberman, H., Fry, C., and Weitzman, L. 2001. Exploring the Web with reconnaissance agents. Commun. ACM 44(8)

Appendix: Sample Run

Note: Light editing of the transcript was done here for presentation purposes.

```
On URL http://webpages.charter.net/allanms/ to depth 2
 ave out degree: 25, ave in degree: 5, pendants: 573
 max out degree: 447, nodes: 1926, max in degree: 99

------------ Weights: link, hub -----------
Step 0: hub diff 1.614710, auth diff 13.541273
Step 1: hub diff 0.043349, auth diff 0.348605
[...]
Step 12: hub diff 0.035227, auth diff 0.317873
Step 13: hub diff 0.063127, auth diff 0.444664
hub score 0.030631, authority score 0.099806
------- HUBS picked ---------
0.131803864891 salehphp8.persianblog.com/
0.108541670395 www.cyberplaces.com/book/chaptr13/ch13.htm
0.108541670395 xml.coverpages.org/kraus-datamodeling-markup.html
0.108541670395 www.lynnareachamber.com/business_resource_center/ecom
0.108541670395 www.lib.utk.edu/refs/business/ecommerce.html
0.108541670395 www.workforcedevelopmentgroup.com/diversity_links.html
0.108541670395 www.internetnews.com/ec-news/article.php/29261
0.108541670395 logic.stanford.edu/people/genesereth/
0.108541670395 www.sigcomm.org/sigcomm95/workshop/attendee.html
0.108541670395 xml.coverpages.org/ecommerceReg.html
0.108541670395 www.slalegal.org/Newsletter/toc_winter1999.htm
0.108541670395 www.fenichel.com/TownHall.shtml
0.108541670395 www.hight.net/habitat/wwwpoint.htm
0.107998444335 www.sc.edu/beaufort/library/pages/links/business.shtml
0.107998444335 news.com.com/Trying+to+regulate+the+global+market/2009
0.107998444335 www.hypernews.org/~liberte/computing/agents.html
0.107998444335 www.well.com/user/leane/
0.107998444335 www.mahoroba.ne.jp/~felix/CF/1999/0201.html
0.107998444335 www.kelley.iu.edu/Retail/search_engines.htm
0.107998444335 www.fulco.lib.in.us/Internet_Links/business.htm
---- AUTHORITIES picked -----
0.221490188819 www.commerce.net
0.109456361013 www.blogger.com
0.0998060421709 webpages.charter.net/allanms/
0.0892383984411 nih.blogspot.com
0.0796862893508 www.wsanchez.net/blog/
0.0783917963947 feeds.feedburner.com/Recondite
0.0769448329573 www.livejournal.com/users/vesper2000/data/rss
0.0769448329573 www.photo.net/bboard/forum-rss?topic_id=1548
0.0769448329573 blogs.netapp.com/dave/?flavor=atom
0.0769448329573 theaudiocritic.com/blog/index.php?blogId=1
0.0769448329573 www.newslettersbyrss.com/92.nbrss
0.0769448329573 www.unstruct.org/?feed=atom
0.0769448329573 e-scribe.com/news/rss
0.0769448329573 www.etherfarm.com/synapse
0.0769448329573 groups.yahoo.com/group/sfbx/
 [...]
```

Run parameters
Graph statistics of base set

Both hub and link weights

Lines elided in editing

Scores for target node

"Picked" means: adjacent equal domains suppressed

Hub scores at left

Authority scores at left

Four lines elided in editing

Link weights only

```
------------ Weights: link ----------
Step 0: hub diff 82.182786, auth diff 13.541273
Step 1: hub diff 0.865077, auth diff 0.370624
[...]
Step 12: hub diff 2.143306, auth diff 0.460679
Step 13: hub diff 1.982433, auth diff 0.431613
hubScore 0.001167, authScore 0.128184
------- HUBS picked ---------
0.116620708189 www.majid.info/mylos/weblog/2004/05/17-1.html
0.0925262180165 ifindkarma.typepad.com/relax/weblogs/index.html
0.0746907863258 profile.typekey.com/ifindkarma/
0.0684172540465 www.majid.info/mylos/weblog/categories/food/index.html
0.0146245919081 ifindkarma.typepad.com/about.html
0.00504767552665 www.aaronsw.com/weblog/
0.00502348401612 www.majid.info/mylos/weblog/categories/photo/index
0.00380309375519 www.ventureblog.com/
0.0037790137254 ianmurdock.com/?cat=3
0.00350100036807 divedi.blogspot.com/
0.00348772674169 www.docuverse.com/blog/donpark/
0.00346825024732 www.mnot.net/blog/
0.00328806442898 educatedguesswork.org/
0.00322684057881 nih.blogspot.com/2004_11_01_nih_archive.html
0.0031560761545 www.camworld.com/
0.00309608105532 www.rtfm.com/movabletype
0.00307617358789 bitsplitter.net/blog/
0.00298473022623 www.educatedguesswork.org/
0.00298470576043 www.rtfm.com/movabletype/
0.00290427916909 webpages.charter.net/allanms/archive/2004_06_01_arch
---- AUTHORITIES picked -----
0.12818412603 webpages.charter.net/allanms/
0.122660351855 nih.blogspot.com
0.117148059216 www.wsanchez.net/blog/
0.0790936091731 feeds.feedburner.com/Recondite
0.0790407177057 blogs.sun.com/roller/rss/tpm
0.0790407177057 www.sa-cd.net/
0.0790407177057 avondale.typepad.com/rawformat/
0.0790407177057 www.python.org/dev/summary/
0.0790407177057 www.christianlindholm.com/christianlindholm/
0.0790407177057 rentzsch.com/rss.xml/
0.0790407177057 blogs.sun.com/roller/page/tucker
0.0790407177057 www.vmunix.com/mark/blog/feed/
0.0790407177057 www.kefta.com/
0.0790407177057 feeds.feedburner.com/laughingsquid
0.0790407177057 blogs.sun.com/roller/page/msw
0.0790407177057 feeds.gawker.com/lifehacker/full
0.0790407177057 www.patentauthority.com
0.0790407177057 feeds.feedburner.com/ventureblog
0.0790407177057 blogs.sun.com/roller/rss/eschrock
0.0790407177057 asack.typepad.com/a_sack_of_seattle/
```

Hub weights only

```
------------ Weights: hub -----------
Step 0: hub diff 2.057673, auth diff 13.950085
Step 1: hub diff 0.052086, auth diff 0.338645
[...]
Step 12: hub diff 0.043959, auth diff 0.338469
Step 13: hub diff 0.061604, auth diff 0.426916
hubScore 0.033063, authScore 0.134776
------- HUBS picked ---------
0.106211802285 salehphp8.persianblog.com/
0.0897658856096 www.contraversion.com/
0.0897658856096 www.evhead.com/
0.0768737767477 spencerwatson.blogspot.com/2005_07_01_spencerwatson
0.0767078392202 evhead.com/2006/03/attention-full.asp
0.0767078392202 goldtoe.net/2006/03/hear-that-whistle-blowin.html
0.0767078392202 www.chinaherald.net/
0.0765666775285 codinginparadise.org/weblog/
0.0765533828501 calhounscannon.blogspot.com/
0.0765533828501 boredgamegeeks.blogspot.com/
0.0765533828501 burningspearmessage.blogspot.com/
0.0765533828501 minimsft.blogspot.com
0.0765533828501 korrespondence.blogspot.com/
0.0765533828501 ludricious.blogspot.com/
0.0765533828501 waveletmonkey.blogspot.com/
0.0765533828501 peteryared.blogspot.com/
0.0690839980251 codinginparadise.org/weblog/2006/01/ajaxian-site-com
0.0684770747983 www.cyberplaces.com/book/chaptr13/ch13.htm
0.0684770747983 xml.coverpages.org/kraus-datamodeling-markup.html
0.0684770747983 www.lynnareachamber.com/business_resource_center/ecommerce.ht
---- AUTHORITIES picked -----
0.136446684177 webpages.charter.net/allanms/#custard-pao
0.134306427744 www.blogger.com/
0.130752552122 www.commerce.net
0.0842050828388 nih.blogspot.com
0.0777977785827 www.blogger.com/app/post.pyra?blogID=3690933&postID=1
0.0726167455814 www.majid.info/mylos/weblog/categories/food/index.html
0.0710555027515 www.wsanchez.net/blog/
0.0699669019887 feeds.feedburner.com/Recondite
0.0684784385069 blogs.sun.com/roller/rss/tpm
0.0684784385069 www.majid.info/mylos/stories/2003/06/18/mylos.html
0.0684784385069 blogs.sun.com/roller/page/msw
0.0684784385069 feeds.gawker.com/lifehacker/full
0.0684784385069 www.patentauthority.com
0.0684784385069 feeds.feedburner.com/ventureblog
0.0684784385069 blogs.sun.com/roller/rss/eschrock
0.0684784385069 asack.typepad.com/a_sack_of_seattle/
0.0684784385069 blogs.sun.com/roller/rss/Gregp
0.0684784385069 www.majid.info/mylos/temboz.html
0.0684784385069 www.livejournal.com/users/vesper2000/data/rss
0.0684784385069 www.photo.net/bboard/forum-rss?topic_id=1548
```

No weights (straight HITS)

```
------------ Weights: (none)  -----------
Step 0: hub diff 91.144047, auth diff 13.950085
Step 1: hub diff 0.961155, auth diff 0.393943
Step 2: hub diff 2.680810, auth diff 0.439840
[...]
Step 12: hub diff 2.398033, auth diff 0.483841
Step 13: hub diff 2.180778, auth diff 0.443533
hubScore 0.001022, authScore 0.120583

------- HUBS picked ---------
0.110774394162 www.majid.info/mylos/weblog/2004/05/17.html
0.0961560195917 ifindkarma.typepad.com/relax/weblogs/index.html
0.0697192801476 www.majid.info/mylos/weblog/categories/food/index.html
0.066138272186 profile.typekey.com/ifindkarma/
0.0590971689103 www.majid.info/mylos/weblog/categories/python/calendar
0.0136133968341 ifindkarma.typepad.com/about.html
0.0109231858199 www.majid.info/mylos/weblog/categories/it/index.html
0.00624161737872 webpages.charter.net/allanms/archive/2004_06_01_arch
0.00557300490741 www.majid.info/mylos/weblog/index.html
0.00435339198998 www.aaronsw.com/weblog/
0.00366981422081 divedi.blogspot.com/
0.00366097541295 webpages.charter.net/allanms/2004/06/shall-i-cough-
0.00359086109554 nih.blogspot.com
0.00331739036419 www.ventureblog.com/
0.00324416963194 ianmurdock.com/?cat=3
0.00309719894632 www.docuverse.com/blog/donpark/
0.00299354741303 www.mnot.net/blog/
0.00295018790754 www.camworld.com/
0.00293891896229 www.rtfm.com/movabletype
0.00279667822292 educatedguesswork.org/
---- AUTHORITIES picked -----
0.120582568297 webpages.charter.net/allanms/
0.114251573071 nih.blogspot.com
0.109066905125 www.wsanchez.net/blog/
0.0708038505455 www.majid.info/mylos/weblog/categories/food/index.html
0.0704626075916 feeds.feedburner.com/Recondite
0.0704186979185 blogs.sun.com/roller/rss/tpm
0.0704186979185 www.majid.info/mylos/stories/2003/06/18/mylos.html
0.0704186979185 blogs.sun.com/roller/page/msw
0.0704186979185 www.sa-cd.net/
0.0704186979185 feeds.gawker.com/lifehacker/full
0.0704186979185 asack.typepad.com/a_sack_of_seattle/
0.0704186979185 blogs.sun.com/roller/rss/Gregp
0.0704186979185 www.majid.info/mylos/temboz.html
0.0704186979185 www.livejournal.com/users/vesper2000/data/rss
0.0704186979185 www.photo.net/bboard/forum-rss?topic_id=1548
0.0704186979185 blogs.netapp.com/dave/?flavor=atom
0.0704186979185 www.microsite.reuters.com/rss/scienceNews
0.0704186979185 theaudiocritic.com/blog/index.php?blogId=1
0.0704186979185 www.fujifilm.co.uk/presscentre/news/index.php
0.0704186979185 blogs.sun.com/roller/rss/sch
```

Top 25 pages in base set ranked by Hamming distance

```
-------- Hamming Distances ---------
108 http://webpages.charter.net/allanms/archive/2005_11_01_archive
117 http://www.wsanchez.net/blog/
122 http://nih.blogspot.com
123 http://webpages.charter.net/allanms/resume.doc
123 http://webpages.charter.net/allanms/popl84.pdf
123 http://nih.blogspot.com/
124 http://blogs.sun.com/roller/rss/alanc
124 http://feeds.feedburner.com/laughingsquid
124 http://writersblocklive.com/wp/wp-atom.php
124 http://thenewnormal.com/index.php/newnormal/rss_atom/
124 http://www.photographyblog.com/index.php/weblog/index/
124 http://feeds.feedburner.com/FractalsOfChange
124 http://feeds.gawker.com/lifehacker/full
124 http://www.etherfarm.com/synapse/rss_atom/
124 http://blogs.sun.com/roller/rss/Gregp
124 http://blogs.sun.com/roller/rss/eschrock
124 http://www.sa-cd.net/
124 http://www.unstruct.org/?feed=atom
124 http://www.photo.net/bboard/forum-rss?topic_id=1548
124 http://theaudiocritic.com/blog/index.php?blogId=1
124 http://cgi.pbs.org/cgi-registry/cringely/cringelyrdf.pl
124 http://blogs.netapp.com/dave/
124 http://www.photographyblog.com/index.php/weblog/rss_atom/
124 http://asseptic.org/blog/rss.php?version=20
124 http://blogs.netapp.com/dave/?flavor=atom
```

A Note on Project Archives

We include with this project report, two other files, organized as separate archives. Refer to the `readme.txt` file associated with each archive to get a list of files within each. One archive is the collection of software developed for this project, the other archive is the set of batch runs analyzed in the Results section.

A third archive, the "big crawl" itsy pagenode database, can be made available on request. It is over 100Mbytes compressed, but is recommended if the reader wishes to reproduce the reported results.