Link-based Approaches for Text Retrieval

Julien Gevrey and Stefan M Rüger
Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London SW7 2BZ, England
gevreyjulien@hotmail.com and s.rueger@doc.ic.ac.uk

Abstract. We assess a family of ranking mechanisms for search engines based on linkage analysis using a carefully engineered subset of the World Wide Web, `WT10g` (Bailey, Craswell and Hawking 2001), and a set of relevance judgements for 50 different queries from Trec-9 to evaluate the performance of several link-based ranking techniques.

Among these link-based algorithms, Kleinberg's `HITS` and Larry Page and Sergey Brin's `PageRank` are assessed. Link analysis seems to yield poor results in Trec's Web Ad Hoc Task. We suggest some alternative algorithms which reuse both text-based search similarity measures and linkage analysis. Although these algorithms yield better results, improving text-only search recall-precision curves in the Web Ad Hoc Task remains elusive; only a certain category of queries seems to benefit from linkage analysis. Among these queries, homepage searches may be good candidates.

1 HITS

"*What other web pages find useful is likely to be useful to me as well.*" This approach is also known as Kleinberg's method (Kleinberg 1998; Chakrabarti, Dom, Gibson, Kleinberg, Raghavan and Rajagopalan 1998; Kleinberg, Kumar, Raghavan, Rajagopalan and Tomkins 1999) and differs drastically from traditional search engines whose ranking method is mainly based on the frequency of matching words. Documents that are pointed to by many other documents are called *authorities* and are ranked highly. Documents that point to many documents related to the query-topic are called *hubs* and may also be of interest. Good authorities are those that are pointed to by good hubs and good hubs point to good authorities: this is the mutually reinforcing relationship at the core of Kleinberg's `HITS` algorithm. `HITS` comprises two phases (Kleinberg 1998):

Phase 1 provides a small subgraph G of the web that is relatively focused on the query topic —it has many relevant pages, and strong authorities. This is done by taking the most highly ranked items of a text-based search (say, the top 200) as root set. This set is then expanded by adding, for each page in the root set, all its incoming links and a fraction of its out-going links. The set of pages thus derived is known as the base set.

Phase 2 consists of analysing the link structure of G to compute hubs and authorities in a mutually reinforcing relationship. This calculation is an iterative process. At step n, the hub value of a page is computed as the sum of the authority values (computed at step $n-1$) of its incoming links, and the authority value of a page is computed as the sum of the hub values (computed at step $n-1$) of its incoming links, ie,

$$a_{n+1}(u) = \sum_{(v,u)\in G} h_n(v)$$

and

$$h_{n+1}(u) = \sum_{(v,u)\in G} a_n(v),$$

where $a_n(u)$ denotes the authority value computed for page u at step n, $h_n(u)$ denotes the hub value computed for page u at step n and G denotes the graph whose nodes are web pages and whose edges are links from one page to another: $(v,u) \in G$ iff page v points to u.

2 PageRank

Larry Page and Sergey Brin's algorithm (1998, see also Page, Brin, Motwani and Winograd 1998) is said to be deployed in their successful search-engine `Google` (`http://www.google.com`). Like Kleinberg's algorithm, it focuses on the hyperlink structure of web pages.

Intuitively, we solve the recursive definition of authority: a page is authoritative if authoritative pages link to it. At each step of the recursion, a page v with outdegree N_v and authority value a_v pointing to some page u will confer on page u a fraction a_v/N_v of its own authority value. For each page in the collection,

$$a_{n+1}(u) = \sum_{(v,u)\in G} \frac{a_n(v)}{N_v},$$

where $a_n(u)$ denotes the authority value computed for page u at step n and G denotes the graph made from links between pages.

Like `HITS`, `PageRank` relies on an eigenvector calculation: eventually, the importance of each page reaches a limit, which happens to be its component in the principal eigenvector of the matrix corresponding to the above transform (the `HITS` matrix is the adjacency matrix of G, whereas the PageRank matrix is a weighted adjacency matrix where 1's are replaced by $1/N_v$'s).

3 Average and Sim

To add other link-based ranking approaches to our experiments we also suggest our own link-based algorithms. These are not fixed-point algorithms like `HITS` and `PageRank`, since they only comprise a single iteration. Our belief is that if the computation of authorities only takes a few steps, the information about the initial ranking is neither lost nor diluted by too much iteration. Furthermore, we want to allow pages to confer some authority on pages they point to.

The underlying idea of our algorithms is to combine the similarity measures obtained by text-only search with linkage analysis. Indeed, `HITS` and `PageRank` work on a root set (top results of a text-based analysis) and then assign to each page the same authority value for the link analysis to work on (because the fixed-point algorithm converges to the principal eigenvector of a given matrix no matter what the initial vector is). This results in losing part of the information (ranking and similarity measures) obtained via text analysis. Our algorithms, called `Average` and `Sim`, reuse similarity measures along with linkage analysis.

Average. The authority value of page p is the average over similarity measures of all incoming links:

$$\text{authority}(p) = \frac{1}{|\{q|q \to p\}|} \sum_{q \to p} \text{similarity}(q)$$

By assigning to p the average value of similarity measures of all pages pointing to p, we mean that we believe more in what others say about p than in what p tells about itself. We use the average instead of a simple sum, because our experiments with Trec-9 Web Track queries have shown that average performs far better. This may be due to the fact that the average lays more emphasis on the quality than on the quantity of incoming links as summing would do. Another advantage of computing the average is that results nearly become root-set size independent.

Sim. The authority value of page p is the similarity measure of page p plus the average over all similarity measures of incoming links (similarity value + authority value conferred by pages pointing to p):

$$\text{authority}(p) = \text{similarity}(p) + \frac{1}{|\{q|q \to p\}|} \sum_{q \to p} \text{similarity}(q)$$

The idea is much the same as what is presented for `Average`. With `Sim`, we take into account both what the page tells us about itself and what other pages have to say. The formula can be decomposed as follows: one term (similarity measure) for text-analysis of page p, one term (average similarity measure over all incoming links) for the authority that other pages confer to p. Owing to the use of the average, both terms are of the same order so that summing up these two terms makes sense.

4 Results Table

Table 1 presents the average precision of several algorithms for the fifty Trec-9 queries. The text-only baseline for linkage analysis is that of Managing Gigabytes: `MG` (Witten, Moffat and Bell 1999).

From these experiments with Trec-9 data, we have shown that linkage analysis does not improve the average precision of content-only search. The influence of several factors on linkage analysis has been studied carefully. We have shown that, for `HITS` and `PageRank` algorithms, the root set should be of medium size (500 pages or so), so that it contains enough relevant pages and not too many irrelevant items. As for the number of iterations, experiments show that it is not necessary to make `HITS` and `PageRank` converge; the best results are obtained with only a few iterations. A new approach that may be of interest would be to try to combine linkage analysis with the reuse of similarity measures, since these methods yield better average results than `HITS` and `PageRank`.

These first unsuccessful attempts to improve text-only search results by linkage analysis in Trec's Web Ad Hoc Task correspond to other results, such as of Singhal and Kaszkiel (2000), Gurrin and Smeaton (2000b), and Savoy and Rasolofo (2000). They also find disappointing

algorithm	root set size	number of iterations	root set expansion	reuse of similarity measures	average precision
HITS	200	40	yes	no	0.0115
HITS	500	40	yes	no	0.0055
HITS	1000	40	yes	no	0.0057
HITS	200	40	no	no	0.0276
HITS	500	40	no	no	0.0216
HITS	1000	40	no	no	0.0215
PageRank	300	40	no	no	0.0285
PageRank	400	40	no	no	0.0297
PageRank	600	40	no	no	0.0232
PageRank	1000	40	no	no	0.0214
PageRank	1000	2	no	no	0.0278
PageRank	1000	10	no	no	0.0299
Average	3000	1	no	yes	0.0506
Sim	3000	1	no	yes	0.0504
MG (baseline)	-	-	-	-	0.0770

Table 1: Link Analysis Results for the previous Trec-9 queries

average precisions for link-based methods. For instance, Gurrin and Smeaton (2000b) have carried out link-based experiments that made a distinction between structural links (that separate documents within a particular domain, exist to aid the user in navigating within a domain, and consequently are not seen as a source of authority judgements) and functional links (that link documents in different domains, and can be seen mostly as links from a source document to a target document that contains similar and, to the author's opinion, useful information). More emphasis was laid on functional links since they are thought to be the most authoritative ones. What Gurrin and Smeaton found in (2000a) and (2000b) is that their link-based method did not bring any improvement over text-only search. However, their conclusion is that these poor results are not necessarily due to their method. As they point out, the WT10g dataset may not be relevant to assess search-engines' performance. Indeed, they found while experimenting on WT10g that, in all, approximately 2% of the links were functional, while a large 98% were structural links, so that the lack of functional links seriously hampers their experiments. Other criticisms have been uttered by Google's co-founders Larry Page and Sergey Brin, who write in (1998) that they do not believe that using a small set such as WT10g (compared to the real WWW) allows us to evaluate how a search-engine would perform while working on the real World Wide Web.

Since Average and Sim are the link-based methods that yield the best link-based results in our experiments, we decided to run these two algorithms for the Web Ad Hoc Task submission of Trec 2001 (icadhoc1 and icadhoc2). Text-only search results were submitted as baseline in icadhoc3. Table 2 displays the results after evaluation – they are of the same quality and quantity as the ones for previous year's queries in Table 1.

Are there query subsets which are likely to benefit from link-analysis? Our experiments in the Web Ad Hoc Task have shown that some queries are good candidates for link analysis; the

algorithm	average precision
`Average` (icadhoc1)	0.0537
`Sim` (icadhoc2)	0.0458
`MG` (baseline icadhoc3)	0.0883

Table 2: Link Analysis Results for the unseen Trec 2001 queries

retrieval performance for these queries being consistently improved by link-based methods no matter what algorithm was deployed. This may be due to the fact that the underlying hyperlink structure is particularly adapted. In this respect, the study of web communities, as Kleinberg does in (Gibson, Kleinberg and Raghavan 1998), may be a means of determining topics and communities for which link-structure is suitable for link-based retrieval techniques. However on average, given the Trec-9 queries, this improvement fails to show.

5 Home Page Finding

An anchor-text algorithm: Anchor. The underlying idea is to consider web pages as hubs for a given topic. `MG` is used to search an "anchor-text collection" built by indexing the anchor-texts of each page in `WT10g`. The top-ranked pages of the anchor-text collection can be thought of as being good hubs for the query, since they contain many anchor-texts related to the query. This allows one to compute hub values for each page in the collection. The authority value is then computed by summing up hub values of all incoming links:

$$\text{authority}(p) = \sum_{q \to p} \text{hubvalue}(q)$$

Only 17 pages were retrieved before rank 100 using `Anchor`, and the average rank over 17 found homepages is 23.82.

This is a very poor result compared to the 57 homepages found before rank 100 by text-only search (`MG`), and the average rank over these 57 found homepages is 16.47.

Rank Merging. On average, the `Anchor` algorithm performs far worse than text-only search. However, it helps retrieve more efficiently home pages that are ranked poorly (below rank 100) by text-based search. This disjointedness of behaviour is a good reason to suggest a merging of the ranked lists for an overall better result. Indeed, rank merging can improve text-only search in the Home Page Finding Task: with a particular merging heuristics we managed to retrieve 61 homepages with an average rank of 17.59.

To obtain this result, we merged `MG`'s and `Anchor`'s ranking lists as follows: the first seven retrieved items from `MG` and the first seven retrieved items from `Anchor` come first (interleaved), followed by 35 items from `MG` and the rest is completed with items retrieved by `Anchor`. This heuristics is based on the observation that the top few results from `MG` and `Anchor` are often good. The 35 following items are from `MG`'s ranking list since on the whole `MG` performs better than `Anchor` and retrieves home pages before the top-40 ranked pages (if it is to retrieve them before rank 100). The last 51 items come from `Anchor`'s ranking list because `Anchor` may help bring interesting pages into the top 100 that are not retrieved by `MG`.

These experiments show that link-based methods can help improve content-only search in home page finding. However, we have not relied on Anchor only to obtain this result; a rank-merging technique combining both text-based results and anchor-text-based results had to be deployed. Also, the Rank-Merging heuristics was chosen so that it improves the behaviour on previous year's Trec queries. It had to be seen with this year's queries whether this heuristics is generically helpful.

Hence, we submitted the link-based Rank Merging for Trec 2001's Home Page Finding submission (ichp1) and text-only search (MG) results as baseline (ichp2). It turns out that the baseline algorithm fares better in terms of the Trec evaluations than Rank-Merging (average reciprocal rank over 145 topics 0.237 vs 0.208).

6 Conclusion

In the Web Ad Hoc Task, we have shown that HITS and PageRank alone can not improve text-only performance. We have studied the influence of the expansion process, the root set size and the number of iterations on HITS and PageRank. We found that — in the context of Trec's WT10g — expansion is not efficient, that the root set should be of medium size to contain enough relevant documents without having too many irrelevant pages and that a small number of iterations is often better than convergence. We suggested two algorithms, Sim and Average, which combine reuse of similarity measures of text-only search with linkage analysis, and yielded better results than HITS and PageRank. However these results are still below MG's text-only search results.

The Home Page Finding Task's experiments illustrate how anchor-text can be efficiently used to retrieve home pages. The relevance of anchor-text in the Home Page Finding Task may be due to the fact that there is little ambiguity in labelling a link that points to a home page; all one has to do is to name the entity, individual or organisation one wants to refer to. By deploying a rank-merging technique, we have seen some anecdotal (but no general) evidence of improvement through link-analysis.

References

P Bailey, N Craswell and D Hawking (2001). Engineering a multi-purpose test collection for web retrieval experiments. In *Notebook Text Retrieval conf 2001*. NIST.

S Brin and L Page (1998). Anatomy of a large-scale hypertextual web search engine. In *Proc 7th WWW conf.*

S Chakrabarti, B Dom, D Gibson, J Kleinberg, P Raghavan and S. Rajagopalan (1998). Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc 7th WWW conf.*

D Gibson, J Kleinberg and P Raghavan (1998). Inferring web communities from link topology. In *Proc 9th ACM conf on Hypertext and Hypermedia.*

C Gurrin and A F Smeaton (2000a). A connectivity analysis approach to increasing precision in retrieval from hyperlinked documents. In *Proc 8th Text Retrieval conf*, pp 357–366. NIST.

C Gurrin and A F Smeaton (2000b). Dublin city university experiments in connectivity analysis for trec-9. In *Notebook 9th Text Retrieval conf*, pp 190ff. NIST.

J Kleinberg (1998). Authoritative sources in a hyperlinked environment. In *Proc 9th ACM-SIAM Symposium on Discrete Algorithms.*

J Kleinberg, R Kumar, P Raghavan, S Rajagopalan and A S Tomkins (1999). The web as a graph: measurements, models and methods. In *Proc 5th Annual Intl Conf Computing and Combinatorics.*

L Page, S Brin, R Motwani and T Winograd (1998). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford, Santa Barbara. http://www-db.stanford.edu/~backrub/pageranksub.ps.

J Savoy and Y Rasolofo (2000). Report on the trec-9 experiment: Link-based retrieval and distributed collections. In *Notebook 9th Text Retrieval conf*, pp 472ff. NIST.

A Singhal and M Kaszkiel (2000). At&t at trec-9. In *Notebook 9th Text Retrieval conf*, pp 134ff. NIST.

I H Witten, A Moffat and T C Bell (1999). *Managing Gigabytes.* Morgan Kaufmann.

Acknowledgements: This work was partially supported by EPSRC, UK.