Combining text- and link-based retrieval methods for Web IR

Kiduk Yang
School of Information and Library Science
University of North Carolina
Chapel Hill, North Carolina 27599-3360, U.S.A.
yangk@ils.unc.edu

0. Submitted Runs

uncvmss, uncvsmm, uncfsls, uncfslm[1] – WT10g automatic topic relevance task runs

1. Introduction

The characteristics of Web search environment, namely the document characteristics and the searcher behavior on the Web, confound the problems of Information Retrieval (IR). The massive, heterogeneous, dynamic, and distributed Web document collection as well as the unpredictable and less than ideal querying behavior of a typical Web searcher exacerbate conventional IR problems and diminish the effectiveness of retrieval approaches proven in the laboratory conditions of traditional IR. At the same time, the Web is rich with various sources of information that go beyond the contents of documents, such as document characteristics, hyperlinks, Web directories (e.g. Yahoo), and user statistics.

Fusion IR studies have repeatedly shown that combining multiple sources of evidence can improve retrieval performance. Furthermore, the nature of the Web search environment is such that retrieval approaches based on single sources of evidence suffer from weaknesses that can hurt the retrieval performance in certain situations. For example, content-based IR approaches have difficulty dealing with the diversity in vocabulary and quality of web documents, while link-based approaches can suffer from incomplete or noisy link topology. The inadequacies of singular Web IR approaches coupled with the fusion hypothesis (i.e. "fusion is good for IR") make a strong argument for combining multiple sources of evidence as a potentially advantageous retrieval strategy for Web IR.

Among the various source of evidence on the Web, we focused our TREC-10 efforts on leveraging document text and hyperlinks, and examined the effects of combining result sets as well as those of various evidence source parameters.

2. Text-based Method: VSM

The text-based retrieval component of the experiment was based on a Vector Space Model (VSM) using the SMART length-normalized term weights as implemented in IRIS[2] (Yang & Maglaughlin, 2000).

2.1 Text Processing

IRIS processed documents by first removing HTML tags and punctuation, and then excluding

[1] Submitted runs along with the entire retrieval system were lost due to a machine crash. Results discussed in this paper are based on post-submission data produced by a recreated system. At the time of submission, uncvsms and uncvsmm were vector space model (VSM) runs using short and medium length queries respectively, and uncfsls and uncfslm were VSM and HITS fusion runs using short and medium queries.

[2] IRIS (Interactive Retrieval Information System) is an experimental retrieval system developed in the School of Information and Library Science at the University of North Carolina.

390 high-frequency terms listed in the WAIS default stopwords list as well as "IRIS stopwords,[3]" which were arrived at by examining the inverted index and identifying low frequency terms that appeared to have little value.

After punctuation and stopword removal, IRIS conflated each word by applying the simple plural remover (Frakes & Baeza-Yates, 1992). The simple plural remover was chosen to speed up indexing time and to minimize the overstemming effect of more aggressive stemmers.

2.2 Term Indexes

In addition to body text terms (i.e. terms between <BODY> and </BODY> tags), IRIS extracted header text terms from document titles, meta keyword and description texts, and heading texts (i.e. texts between <Hn> and </Hn> tags). A combination of body and header text terms was also created, where the header text terms were emphasized by multiplying the term frequencies by 10.

In each of the three term sources, adjacent noun phrases were identified to construct noun phrase indexes as well as single term indexes. By using an online dictionary and the punctuation-based phrase window recognition algorithm, IRIS defined an adjacent noun phrase as consisting of up to three adjacent dictionary nouns or capitalized words within a phrase window.

2.3 Document Ranking and Pseudo-feedback

Documents were ranked in decreasing order of the inner product of document and query vectors,

$$\mathbf{q}^{\mathrm{T}}\mathbf{d}_i = \sum_{k=1}^{t} q_k d_{ik} \,, \qquad (1)$$

where q_k is the weight of term k in the query, d_{ik} is the weight of term k in document i, and t is the number of terms in the index. SMART *Lnu* weights with the slope of 0.3 were used for document terms (Buckley et al., 1996; Buckley et al., 1997), and SMART *ltc* weights (Buckley et al., 1995) were used for query terms. *Lnu* weights attempt to match the probability of retrieval given a document length with the probability of relevance given that length (Singhal et al., 1996).

Top ten positive and top two negative weighted terms from the top three ranked documents of the initial retrieval results were used to expand the initial query in a pseudo-feedback retrieval process.

2.4 VSM systems

Table 1 enumerates the text-based method parameters for VSM systems, which are query length, term source, use of phrase terms, and use of pseudo-feedback. Query length range from short (topic title) and medium (topic title and description) to long (topic title, description, and narrative). Term sources are body text, header text, and body plus header text. The combination of parameters (3 query lengths, 3 term sources, 2 for phrase use, 2 for feedback use) resulted in 36 VSM systems.

[3] IRIS stopwords for TREC-10 Web track experiment were defined as all non-alphabetical words (exception: embedded hyphen), words consisting of more than 25 or less than 3 characters, and words that contain 3 or more repeated characters.

Table 1. VSM system* parameters

Query length	*Term Source*	*Noun Phrase*	*Pseudo-feedback*
short	body text	no	no
medium	header text	yes	yes
long	body + header		

*VSM system name = vsm$qform$index$phrase.$feedback (e.g. vsmsb0.1)
where $qform = query length (s, m, l)
$index = term source (b, h, bh)
$phrase = noun phrase (0, 1)
$feedback = pseudo-feedback (1, 2)

3. Link-based Method: HITS

Among the several possible link-based retrieval methods, the authority scores of documents computed by the HITS algorithm (Kleinberg, 1997) were used to generate a ranked list of documents with respect to a given query. PageRank scores (Page et al., 1998) could be used to rank documents as well, but effectiveness computation of PageRank scores is likely to require a much larger set of linked documents than WT10g corpus (Brin & Page, 1998). The Clever algorithm that extends HITS by incorporating the text around links into the computation of hub and authority scores has been shown to improve the performance of HITS (Chakrabarti et al., 1998). However, Clever combines link- and text-based methods implicitly and thus makes it difficult to isolate the contributions and behaviors of individual methods, which we wanted to study to better understand the effect of combining retrieval result sets.

HITS defines "authority" as a page that is pointed to by many good hubs and defines "hub" as a page that points to many good authorities. Mathematically, these circular definitions can be expressed as follows:

$$a(p) = \sum_{q \to p} h(q), \qquad (2)$$

$$h(p) = \sum_{p \to q} a(q). \qquad (3)$$

The above equations define the authority weight *a(p)* and the hub weight *h(p)* for each page *p*, where *p→q* denote "page *p* has a hyperlink to page *q*".

HITS starts with a root set *S* of text-based search engine results in response to a query about some topic, expands *S* to a base set *T* with the inlinks and outlinks of *S*, eliminates links between pages with the same domain name in *T* to define the graph *G*, runs the iterative algorithm (equations 2 and 3) on *G* until convergence, and returns a set of documents with high *h(p)* weights (i.e. hubs) and another set with high *a(p)* weights (i.e. authorities). The iterative algorithm works as follows: Starting with all weights initialized to 1, each step of the iterative algorithm computes *h(p)* and *a(p)* for every page *p* in *T*, normalizes each of them so that the sum of the squares adds up to 1, and repeats until the weights stabilize. In fact it can be shown that the authority weights at convergence correspond to the principal eigenvalues of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ and hub weights correspond to those of $\mathbf{A}\mathbf{A}^{\mathbf{T}}$, where **A** is the link matrix of the base set *T*[4]. Typically, convergence

[4] The $(i,j)^{th}$ entry of A is 1 if there exists a link from page *i* to page *j*, and is 0 otherwise. In A^T, the transpose of the link matrix A, the $(i,j)^{th}$ entry of A corresponds to the link from page *j* to page *i*. The $(i,j)^{th}$ entry of AA^T gives the number of pages pointed to by both page *i* and page *j* (bibliometric coupling), while the $(i,j)^{th}$ entry of A^TA gives the number of pages that point to both page *i* and page *j* (cocitation).

occurs in 10 to 50 iterations for *T* consisting of about 5000 Web pages, expanded from the root set *S* of 200 pages while being constrained by the expansion limit of 50 inlinks per page.

3.1 Modified HITS Algorithm

The original HITS algorithm was modified by adopting a couple of improvements from other HITS-based approaches. As implemented in the ARC algorithm (Chakrabarti et al., 1998), the root set was expanded by 2 links instead of 1 link (i.e. expand *S* by all pages that are 2 link distance away from *S*). Also, the edge weights by Bharat and Henzinger (1998), which essentially normalize the contribution of authorship by dividing the contribution of each page by the number of pages created by the same author, was used to modify the HITS formulas as follows:

$$a(p) = \sum_{q \to p} h(q) \times auth\_wt(q,p)\,, \qquad (4)$$

$$h(p) = \sum_{p \to q} a(q) \times hub\_wt(p,q)\,. \qquad (5)$$

In above equations, *auth_wt(q,p)* is 1/*m* for page *q*, whose host has *m* documents pointing to *p*, and *hub_wt(p,q)* is 1/*n* for page *q*, which is pointed by *n* documents from the host of *p*.

3.2 Host Definitions

To compute the edge weights of modified HITS algorithm, one must first establish a definition of a host to identify the page authorship (i.e. documents belonging to a given host are created by the same author). Though host identification heuristics employing link analysis might be ideal, we opted for simplistic host definitions based on URL lengths. Short host form was arrived at by truncating the document URL at the first occurrence of a slash mark (i.e. '/'), and long host form from the last occurrence.

3.3 HITS systems

Among the 36 text-based system results, we chose the best performing system with all variations of query lengths. The combination of host definition and seed set parameters, as seen in Table 2 below, resulted in 6 HITS systems.

Table 2. HITS system* parameters

Host Definition	*Seed Set*
short	short query, body text, phrase, no feedback
long	medium query, body text, phrase, no feedback
	long query, body text, phrase, no feedback

*HITS system name = hit$hform$seed (e.g. hitssb1.1)
where $hform = host definition (s, l)
$seed = seed set (sb1.1, mb1.1, lb1.1)

4. Fusion Method

Since it is not clear from literature how much can be gained by using one fusion method over another, the Similarity Merge method (Fox & Shaw, 1993, 1994) was chosen for its simplicity and consideration of overlap, which is thought to be an important factor in fusion. Equation (6) below describes the fusion formula used to merge and rank documents retrieved by different systems:

$$FS = (\sum NS_i)*\text{overlap}, \qquad (6)$$

where: FS = fusion score of a document,
NS_i = normalized score of a document by method i,
overlap = number of methods that retrieved a given document.

The normalized document score, NS_i, is computed by Lee's min-max formula (1996, 1997), where S_i is the retrieval score of a given document and S_{max} and S_{min} are the maximum and minimum document scores by method i.

$$NS_i = (S_i – S_{min}) / (S_{max} – S_{min}), \quad (7)$$

5. Results

Although various fusion combinations were tried, combining retrieval result sets did not improve on the performance of the best text-based method. In fact, fusion in general seemed to decrease retrieval performance, which is contrary to previous fusion research findings that suggest that combining results of various retrieval methods is beneficial to retrieval performance.

Curiously enough, past TREC participants who tried fusion with WT10g corpus also found that combining text- and link-based methods did not improve retrieval performance (Singhal & Kaszkiel, 2001; Gurrin & Smeaton, 2001; Savoy & Rasolofo, 2001). Whether this is simply an artifact of the WT10g test collection (i.e. link structure, relevance judgments, query characteristics) or the reflection of real inadequacies present in link analysis and/or fusion methods remains the main focus in our ongoing investigation.

5.1 Single System Results

The best performing VSM system, measured by average precision of 0.1406, was vsmlb1.1 (long query, body text, noun phrase, no feedback). The best HITS system was hitslb1.1 (short host, seed set system of vsmlb1.1) with average precision of 0.0399. The best text-based system not only outperformed the best link-based system (3.5 times better in average precision), but also outperformed all other systems, both single and fusion, as can be seen in subsequent sections.

Examination of single system results (Table 3) reveals some interesting phenomena regarding the effects of individual system parameters on retrieval performance. According to Table 3, the system parameters most influential to retrieval performance seem to be index source, query length, and host definition. VSM systems using header terms only show markedly worse performance than systems using body text terms, and longer query length systems generally perform better than shorter query systems using the same index source terms. The shorter host definition is obviously far superior to longer definition (over 13 times better in average precision) for HITS systems.

In post analysis, we constructed optimum seed sets from known relevant documents to ascertain the maximum performance level possible by HITS method for WT10g corpus. Although the HITS system with optimum seed set and short host definition resulted in an average precision value eight times that of the best HITS system (0.3144 vs. 0.0399), it is somewhat disappointing as a maximum performance threshold. One could even view this as the failing of HITS algorithm, which reduces the seed system performance by one third at best.

Table 3. Single System Results

VSM systems	*Average Precision*	*HITS systems*	*Average Precision*
vsmlb1.1	**0.1406**	hitsopt[1]	0.3144
vsmlb0.1	0.1387	hitlopt[2]	0.0447
vsmlb0.2	0.1345	**hitslb1.1**	**0.0399**
vsmlb1.2	0.1339	hitsmb1.1	0.0382
vsmmb1.1	0.1272	hitssb1.1	0.0314
vsmmb1.2	0.1254	hitllb1.1	0.0029
vsmmb0.1	0.1247	hitlmb1.1	0.0026
vsmmb0.2	0.1233	hitlsb1.1	0.0008
vsmlbh1.1	0.1148		
vsmlbh0.1	0.1114		
vsmlbh0.2	0.1103		
vsmlbh1.2	0.1079		
vsmsb1.1	0.1054		
vsmsb0.1	0.1038		
vsmsb1.2	0.1036		
vsmsb0.2	0.1032		
vsmmbh1.1	0.1017		
vsmmbh0.1	0.0998		
vsmmbh1.2	0.0988		
vsmmbh0.2	0.0973		
vsmsbh1.1	0.0842		
vsmsbh0.1	0.0830		
vsmsbh1.2	0.0819		
vsmsbh0.2	0.0815		
vsmmh0.1	0.0210		
vsmmh1.1	0.0208		
vsmlh0.2	0.0190		
vsmlh0.1	0.0182		
vsmmh0.2	0.0182		
vsmsh1.1	0.0181		
vsmsh0.1	0.0179		
vsmlh1.2	0.0176		
vsmlh1.1	0.0172		
vsmmh1.2	0.0163		
vsmsh0.2	0.0151		
vsmsh1.2	0.0133		

hitsoptimum[1] = short host, optimum seed set
hitloptimum[2] = long host, optimum seed set

The performance of the optimum HITS system in Table 3 may not necessarily reflect the true potential of link analysis approach. In addition to potential effects of incomplete relevance judgments and truncated link structure with heavy concentration of spurious links in WT10g collection (Gurrin & Smeaton, 2001), we note that 42 out of 50 TREC-10 topics have less than 100 known relevant documents. In fact, 31 of those 42 topics have less than 50 known relevant documents. The topics with small number of relevant documents mean noisy seed sets, even when the perfect results have been achieved by a seed retrieval system (i.e. over three quarters of the seed set of size 200 will consist of irrelevant documents for 31 topics), which are likely to bring in more noise during link expansion and thus result in expanded sets with dominant link structures unrelated to the original topics.

Another point to consider about the HITS method is its tendency to rank documents in relatively small clusters, where each cluster represents mutually reinforcing communities (i.e. hubs and authorities) on sufficiently broad topics. This tendency could rank clusters of non-relevant documents with dense link structure above sparsely linked relevant documents, which will adversely affect average precision but may not affect high precision.

5.2 Fusion System Results

Table 4 and 5 show the fusion performances of combining various VSM and HITS system results. It is interesting to note that the best VSM fusion result (0.1354 in Table 4) is worse than the best VSM single system result (0.1406 in Table 3), while the best HITS fusion result (0.0540 in Table 5) is better than the best HITS single system result (0.0399 in Table 3). One possible explanation for this phenomenon may be that the best VSM system dominates all other systems (i.e. additional relevant documents introduced by other system are negligible), while the best HITS system result is enhance by unique contributions from other HITS systems. In other words, HITS systems may produce more diverse result sets than VMS systems and are thus helped by fusion.

Combining text- and link-based systems (Table 6) resulted in performance degradation of text-based results, even when the best HITS and VSM systems were combined (0.1012 in Table 6 vs. 0.1406 in Table 3). When the optimum HITS result was combined with the best VSM result (0.3144 and 0.1406 in Table 3), however, the improvement by fusion was almost linear (0.4549). Although such fusion system is unrealistic, it does suggest the fusion potential where optimum performance level of one method can be raised by combining it with a reasonably effective method of a different kind.

Table 4. VSM fusion systems

Systems	*Query Length*	*Term Index*	*Pseudo-feedback*	*Average Precision*
fvsmb0.1	all	body text, no phrase	no	0.1331
fvsmb0.2	all	body text, no phrase	yes	0.1297
fvsmb1.1	**all**	**body text, phrase**	**no**	**0.1354**
fvsmb1.2	all	body text, phrase	yes	0.1309
fvsmh0.1	all	header text, no phrase	no	0.0193
fvsmh0.2	all	header text, no phrase	yes	0.0176
fvsmh1.1	all	header text, phrase	no	0.0196
fvsmh1.2	all	header text, phrase	yes	0.0166
fvsmbh0.1	all	body+header, no phrase	no	0.1046
fvsmbh0.2	all	body+header, no phrase	yes	0.1017
fvsmbh1.1	all	body+header, phrase	no	0.1074
fvsmbh1.2	all	body+header, phrase	yes	0.1039
fvsms.1	short	all	no	0.0729
fvsms.2	short	all	yes	0.0697
fvsmm.1	medium	all	no	0.0886
fvsmm.2	medium	all	yes	0.0840
fvsml.1	long	all	no	0.1055
fvsml.2	long	all	yes	0.0979
fvsm.1	all	all	no	0.0956
fvsm.2	all	all	yes	0.0920
fvsm	all	all	all	0.0947

Table 5. HITS fusion systems

Systems	*Host Definition*	*Seed Set*	*Average Precision*
fhitsb11	all	vsmsb1.1	0.0231
fhitmb11	all	vsmmb1.1	0.0303
fhitlb11	all	vsmlb1.1	0.0304
fhits	**short**	**all**	**0.0540**
fhitl	long	all	0.0032
fhit	all	all	0.0407

Table 6. HITS + VSM fusion systems

System Name	HITS	VSM	Average Precision
fhsopt	optimal system (hitsopt)	best system (vsmlb1.1)	0.4549
fhsbest	best system (hitslb1.1)	best system (vsmlb1.1)	0.1012
fhsv.1	all with short host	all with no feedback	0.1017
fhsv.2	**all with short host**	**all with feedback**	**0.1019**
fhlv.1	all with long host	all with no feedback	0.0999
fhlv.2	all with long host	all with feedback	0.1017
fhv.1	all	all with no feedback	0.0999
fhv.2	all	all with feedback	0.1017
fhvs.1	all	all with short query, no feedback	0.0782
fhvs.2	all	all with short query, feedback	0.0963
fhvm.1	all	all with medium query, no feedback	0.0879
fhvm.2	all	all with medium query, feedback	0.0980
fhvl.1	all	all with long query, no feedback	0.0999
fhvl.2	all	all with long query, feedback	0.0999
fhv.1	all	all without feedback	0.1018
fhv.2	all	all with feedback	0.1018
fhv	all	all	0.1018

6. Conclusion

In WT10g topic relevance task, we examined the effect of combining result sets as well as those of various evidence source parameters for text- and link-based methods. Analysis of results suggests that index source, query length, and host definition are the most influential system parameters for retrieval performance. We found link-based systems, HITS in particular, to perform significantly worse than text-bases systems, and combining results sets using the similarity merge formula did not enhance retrieval performance in general. Performance improvement by fusion occurred only on two occasions: once when HITS systems with short host definition were combined, and another time when the optimum HITS result was combined with the best VSM result.

The general failure of fusion evidenced in our results could be due to the characteristics of WT10g test collection, failings of link analysis, inadequacies of fusion formula, or combinations of all or any of the above. The optimum fusion combination result suggests to us that fusion potential exists despite possible shortcomings of the test collection and individual retrieval methods. Consequently, we believe the future fusion efforts should focus on discovering the fusion formula that can best realize the fusion potential of combining diverse retrieval methods.

References

Bharat, K. & Henzinger, M. R. (1998). Improved Algorithms for Topic Distillation in Hyperlinked Environments. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 104-111.

Brin, S. & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the 7th International World Wide Web Conference*, 107-117.

Buckley, C., Salton, G., & Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. In D. K. Harman (Ed.), *The Third Text Rerieval Conference (TREC-3)* (NIST Spec. Publ. 500-225, pp. 1-19). Washington, DC: U.S. Government Printing Office.

Buckley, C., Singhal, A., & Mitra, M. (1997). Using query zoning and correlation within SMART: TREC 5. In E. M. Voorhees & D. K. Harman (Eds.), *The Fifth Text REtrieval Conference (TREC-5)* (NIST Spec. Publ. 500-238, pp. 105-118). Washington, DC: U.S. Government Printing Office.

Buckley, C., Singhal, A., Mitra, M., & Salton, G. (1996). New retrieval approaches using SMART: TREC 4. In D. K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (NIST Spec. Publ. 500-236, pp. 25-48). Washington, DC: U.S. Government Printing Office.

Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., & Kleinberg, J. (1998b). Automatic resource list compilation by analyzing hyperlink structure and associated text. *Proceedings of the 7th International World Wide Web Conference.*

Fox, E. A., & Shaw, J. A. (1994). Combination of multiple searches. In D. K. Harman (Ed.), The Second Text Rerieval Conference (TREC-2) (NIST Spec. Publ. 500-215, pp. 243-252). Washington, DC: U.S. Government Printing Office.

Fox, E. A., & Shaw, J. A. (1995). Combination of multiple searches. In D. K. Harman (Ed.), *The Third Text Rerieval Conference (TREC-3)* (NIST Spec. Publ. 500-225, pp. 105-108). Washington, DC: U.S. Government Printing Office.

Frakes, W. B., & Baeza-Yates, R. (Eds.). (1992). *Information retrieval: Data structures & algorithms*. Englewood Cliffs, NJ: Prentice Hall.

Gurrin, C. & Smeaton, A.F. (2001). Dublin City University experiments in connectivity analysis for TREC-9. In E. M. Voorhees & D. K. Harman (Eds.), *The Nineth Text Rerieval Conference (TREC-9)*. Washington, DC: U.S. Government Printing Office.

Kleinberg, J. (1997). Authoritative sources in a hyperlinked environment. *Proceeding of the 9th ACM-SIAM Symposium on Discrete Algorithms*.

Lee, J. H. (1996). *Combining multiple evidence from different relevance feedback methods (Tech. Rep. No. IR-87).* Amherst: University of Massachusetts, Center for Intelligent Information Retrieval.

Lee, J. H. (1997). Analyses of multiple evidence combination. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 267-276.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The PageRank citation ranking: Bringing order to the Web. *Unpublished*

Savoy, J. & Rasolofo, Y. (2001). Report on the TREC-9 experiment: Link-based retrieval and distributed collections. In E. M. Voorhees & D. K. Harman (Eds.), *The Nineth Text Rerieval Conference (TREC-9)*. Washington, DC: U.S. Government Printing Office.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 21-29.

Singhal, A. & Kaszkiel, M. (2001). AT&T Labs at TREC-9. In E. M. Voorhees & D. K. Harman (Eds.), *The Nineth Text Rerieval Conference (TREC-9)*. Washington, DC: U.S. Government Printing Office.

Yang, K. & Maglaughlin, K. (2000). *IRIS at TREC*-8. In E. M. Voorhees & D. K. Harman (Eds.), *The Eighth Text Rerieval Conference (TREC-8)*. Washington, DC: U.S. Government Printing Office.