ISBN: 972-99353-6-X © 2005 IADIS

MAIN RETRIEVAL SYSTEMS' PARAMETERS ANALYZE

João Ferreira
Instituto Superior de Engenharia de Lisboa
jferreira@deetc.isel.ipl.pt

Alberto Rodrigues da Silva
INESC-ID, Instituto Superior Técnico
alberto.silva@acm.org

José Delgado
Instituto Superior Técnico
Jose.Delgado@tagus.ist.utl.pt

ABSTRACT

We explore retrieval effectiveness of vectorial, link analyses, probabilistic and classification retrieval system parameters and compare the different performance in a controlled environment (using WT10g collection from TREC). Main retrieval parameters are captured from related to: (1) retrieval methods (e.g., vectorial, link analyses, probabilistic and classification); (2) main internal system parameters (e.g., query length, URL length, phrase, feedback, index); (3) and also internal system parameters combination (e.g., combination of different query and URL lengths, phrase, feedback and index) and we concluded about most important parameters, retrieval method performance and that combination can improve results. We analyses many cases from around 500 retrieval systems using our own modular platform, WebSearchTester.

KEYWORDS

Vectorial, Links Analyses, Retrieval Information, Classification, and Probabilistic.

1. INTRODUCTION

How do we find information on the Web? It is an old question far from being solved. Web information is distributed, decentralized and huge in size. The Web can be viewed as one big virtual document collection. The findings from traditional Information Retrieval (IR) research (traditional IR means text-based approaches), however, may not always be applicable in the Web setting. The Web document collection, massive in size and diverse in content, context, format, purpose and quality, challenges the validity of previous research findings based on relatively small and homogeneous test collections. Also, some traditional IR approaches may be applicable in theory, but may not be possible or practical to implement in a Web IR system. For instance, the size, distribution and dynamic nature of information on the Web make it difficult, if not impossible, to construct a complete and up-to-date data representation required for an ideal IR system. In addition, conventional evaluation measures, such as precision, recall, and even relevance, may no longer be applicable to Web IR, where a test collection representative of dynamic and diverse Web data is all but impossible to construct.

To further complicate the matter, information seeking on the Web is quite diverse in characteristics and unpredictable in nature. Web searchers come from all kinds of reasons motivated by all types of information need. The wide range of experience, knowledge, motivation, need, and purpose of Web searchers means that searchers can express wide ranges of information needs in a wide variety of ways with various criteria for satisfying their needs.

At the same time, the Web is rich with new types of information not present in most previous test collections. Hyperlinks, usage statistics, document mark up tags and bodies of topic hierarchies such as Yahoo present an opportunity to leverage the Web-specific document characteristics in novel approaches that go beyond the term-based retrieval framework of traditional IR.

This paper explores the question *of identified the most important parameters of the methods: link analysis, content analysis, and classification-based,* is divided in 4 sections. Firstly, we introduce the problem, secondly we describe individual retrieval systems. Third section we represent the results and on section four the conclusion.

2. RETRIEVAL SYSTEMS

To control and have measures of systems performance we use an controlled environment provide by TREC. As the source for these experiences we use the WT10g collection [1], which is a ten-gigabyte subset of the 1997 Web crawl by the Internet Archive, consists of 1.7 million Web documents, 100 TREC queries (topics 451-550), and official NIST relevance judgments. The WT10g collection also includes the connectivity data, which provides lists of inlinks and outlinks of all documents in the collection.

As classification systems for the Web lack an ideal Web directory, we use Yahoo <http://yahoo.com> due to its size and popularity. Yahoo is the largest and the most widely used Web directory, and consists of 14 top categories with over 645,000 sub-categories that contain almost 3 million Web pages, which are classified and annotated by over 150 professional Yahoo cataloguers.

VSM

The text-based retrieval component is based on a Vector Space Model (VSM) using the SMART length-normalized term weights as implemented in OpenFts <http://openfts.sourceforge.net/>. For implementation details, see [2]. From text, tags are removed; stop words and weights are based on *Lnu* document term weight (1):

$$d_{ik} = \frac{(1+\log(f_{ik}))\big/(1+\log(avg\_f_i))}{(1.0-slope)*p_i + slope*t} \qquad (1)$$

with the slope of 0.3 for document terms [3], where f_{ik} is the number of times term k appears in document i; avg_f_i is the average in-document frequency for document i; t is the number of unique terms in the collection and p_i is the average number of unique terms in a document i. The formula for *ltc* query term weight is:

$$q_k = \frac{(\log(f_k)+1)*idf_k}{\sqrt{\sum_{j=1}^{t}\left[(\log(f_j)+1)*idf_j\right]^2}} \qquad (2)$$

where f_k is the number of times term k appears in the query; and idf_k is the inverse document frequency [4] of term k. The denominator is a document length normalization factor, which compensates for the length variation in queries. Documents were ranked in decreasing order of the inner product of document and query vectors,

$$\mathbf{q}^{\mathrm{T}}\mathbf{d}_i = \sum_{k=1}^{t} q_k d_{ik} \qquad (3)$$

For feedback, we use the top ten positive and top two negative weighted terms from the top three ranked documents of the initial retrieval results. These terms were used to expand the initial query in a pseudo-feedback retrieval process based on the adaptive linear model. The basic approach of the adaptive linear model, which is based on the concept of preference relations from decision theory [5], is to find a solution vector that will rank a more-preferred document before a less-preferred one [6]. The solution vector is arrived at via an error-correction procedure, which begins with a starting vector $\mathbf{q}_{(0)}$ and repeats the cycle of "error-correction" until a vector is found that ranks documents according to the preference order estimation based on relevance feedback [7]. The error-correction cycle i is defined by

$$\mathbf{q}_{(i+1)} = \mathbf{q}_{(i)} + \alpha\mathbf{b} \qquad (4)$$

where α is a constant, and $\mathbf{b}$ is the *difference vector* resulting from subtracting a less-preferred document vector from a more preferred one [8].

We tested 36 VSM based on the combination of four parameters (notation;p/m/l;c/t/d;0/1;0/1): query length (small(p), medium(m), large(l)), term sources (body (c), header (t), document all (d)), phrase use (1-yes;0-no) and feedback use (1-yes;0-no).

Table 1. Notation used for VSM retrieval system. (v$query_lenght$index$phrases$feedback)

Sistem	*Query Lenght*	*Index*	*Phrases*	*Feedback*
*v * * * **	*p-short*	*d –complete document*	*0- without*	*0- no*
	m-medium	*c- body document*	*1- with*	*1 - yes*
	l-long	*t-header document*	*F- combination*	*F- combination*
	F-combination	*F-combination*		

Link analyses

The HITS system's algorithm was modified by adopting a couple of improvements from other HITS-based approaches. As implemented in the ARC algorithm [9], 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*). All intrahost links and stoplist URLs were eliminated from the hub and authority score computations. Stoplist URLs, defined as Web pages with very high indegree, were selected from the list of URLs with indegree greater than 500. Also, the edge weights by [9], 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 formulae as follows:

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

$$h(p) = \sum_{p \to q} a(q) \times hub\_wt(p,q) \qquad (6)$$

In the formulae above, *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 to by *n* documents from the host of *p*. To compute the edge weights of the modified HITS algorithm as well as to eliminate intrahost links, 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 latest occurrence. We tested 6 Hits systems based on 2 parameters: host definition (short (p), long (l)); seed set from VSM systems ((p) Vpc10, (m)Vmc10, (l)Vlc10).

Table 2. Notation used for HITS retrieval system. (h$seed_set$site_length)

Sistem	*Seed set (v*c10)*	*Site length*
*h * **	*short (p)*	*short (p)*
	medium (m)	*long (l)*
	Long (l)	*combination (F)*
	combination (F)	

Classification

The Web Directory search was implemented based on the Term Match (TM) method. TM takes a simpler approach of finding categories in which query terms occur by extending the typical category search implementation of Web directory services.

The first phase of the TM method, which produces a ranked list of categories for a query, matches query terms to terms in the Yahoo sitemap files (i.e. category labels, Yahoo site titles and descriptions, URLs) to find a set of matching nodes in the classification hierarchy and generates a ranked category list in the following manner:

1. For each matching category, (i) compute *tfc* (number of unique query terms in the category label); (ii) compute *tfs* (number of unique query terms in the site title and description) in all its sites; (iii) compute *pms* (proportion of sites with query terms in the category).
2. Rank the matching categories in the descending order of *tfc*, *tfs*, and *pms*.

Note that categories ranked via sorting by multiple variables in such an order that the terms in category labels, which are likely to be highly "powerful", are given precedence over terms in site titles or descriptions. This ranking approach is similar to how Yahoo ranks its search results except that it combines the category and site match results while collapsing the site match results to their parent categories.

The second phase of the TM method is to expand query vector (the class centroid in the TM method) that is built from the best matching categories to produce a ranked list of the WT10g documents. The expanded query vector of the TM method is a vector of selected category terms with normalized term-category association weights. The parameters tested for the TM systems are the number of top categories used, the WT10g term index and terms for pseudo-feedback. The combination of the parameters (3 top categories (1/2/3), 4 WT10g term index (body text, no phrase (*1*) body text, phrase (2) body+header, no phrase (3) body+header, phrase (4), 2 for feedback use(1-yes,0-no)) resulted in 24 TM systems.

Table 3. Notation used for classification based retrieval system.

Sistema	*# Top Cat*	*Index*		*Feedback*
*t * * **	*1*	*body doc no phrase (1)*	*body doc with phrase (2)*	*0*
	2	*header doc no phrase (3)*	*header doc with phrase (4)*	*1*
	3	*F*	*F*	*F*
	F			
tm$# top cat$index$feedback				

Probabilistic

The probabilistic approach involves a two-step process of contingency table construction and association weight calculation. If each document $\boldsymbol{D_i}$ in a collection is regarded as a multi-set a_i of m document terms and b_j of n query terms, i.e. $\boldsymbol{D_i} = (\{a_{i1},\ldots,a_{im}\};\{b_{j1},\ldots,b_{jn}\})$, the associations contained in a particular document $\boldsymbol{D_i}$ consists of all the ordered pairs that can be formed from a_{im}X b_{jn} document subparts. For each term A and a category B (i.e. a_{im}-b_{jn} pair), a contingency table is formed containing the counts for each of the possible combinations of A and B:

AB	A¬B
¬AB	*¬A¬B*

where "¬" denotes the absence of some event. The possible combinations are *AB*, where the events both occur; *A¬B*, where event *A* occurs without *B*; *¬AB*, where *B* occurs without *A*; and finally, *¬A¬B* where neither *A* nor *B* occurs. As each of the pairs for each document is considered, contingency tables are constructed and updated. When all the pairs and contingency tables have been recorded after processing all the documents in a collection, the strength of the associations can be computed for each document_term-query_term pair using a likelihood ratio statistic as the measure of association. The strength of association is computed by the following formula:

$$\lambda' = 2\left[\ln\frac{L(p_1,k_1,n_1)}{L(p,k_1,n_1)} + \ln\frac{L(p_2,k_2,n_2)}{L(p,k_2,n_2)}\right], \quad (7) \quad \text{where: } \ln\left(L(p,k,n)\right) = k\ln p + (n-k)\ln(1-p)\,;\ p_1 = \frac{k_1}{n_1}\,;$$

$$p_2 = \frac{k_2}{n_2}\,;\ p = \frac{k_1+k_2}{n_1+n_2}\,;\ k_1 = AB,\ \ n_1 = AB{+}\neg AB,\ \ k_2 = A\neg B,\ and\ n_2 = A\neg B{+}\neg A\neg B.$$

For each entry document_term-query_term pair, the strength of association is calculated. Documents are order by the some of all document terms associated with query terms like: document_*term*$_1$-query_term$_1$ with weights w_1 and document_*term*$_1$-query_term$_2$ with weights w_2 can be collapsed into *category*$_1$ with weights (w_1+ w_2).

Table 4. Notation used for Probabilistic retrieval system.

sistem	*Index WT10g (Phrases)*	*Query length*	*feedback*
P * **	*header (t)*	*short (p)*	*no (1)*
	document (d)	*medium (m)*	*yes (1)*
	F	*long (l)*	*F*
		Combination (F)	

3. RESULTS

<u>Individual systems:</u> Among the VSM System parameters tested, which are query length, term source, use of phrase terms, and use of pseudo-feedback, query length and term source were found to be most influential to the retrieval outcome. The influence of query length, which may be related to the amount of information, seems intuitive. Regardless of other parameter combinations, longer queries performed better than shorter queries in all cases except when system performances were degraded by the adverse effect of header text terms. The system performance order with regard to query length and term source can be written as:

vlc* > vmc* > vld* > vpc* > vmd* > vpd* > v*t* , (8)

The adverse effect of header text terms (i.e. HTML titles, meta keywords and descriptions, headings text marked up by <H> tags) appear even more pronounced when results are grouped by term source. All body text systems (v*c*) performed better than ones using complete document (v*d*) except when shorter queries were used (i.e. vld*/vmd* > vpc*). The severe degradation of performance introduced by the use of header text terms can be seen in Figure 1, where the performance differentials between the top body text and the top header text systems by all measures are quite pronounced. It is somewhat surprising that header text had such an adverse effect on retrieval performance. Ideally, document titles and headings, not to mention meta description and keywords should contain important concepts of the document, and thus be beneficial for retrieval. The fact that the results show otherwise could be due to the nature of Web documents, which may sometimes be intentionally misleading or constructed with less attention to content and more attention to appearance than purely textual documents.

The use of phrases resulted in only a marginal increase in performance. Similarly, the use of pseudo-feedback resulted in a slight decrease in performance in most cases. In comparison with TREC official runs, the best VSM systems for short and long queries fall between the third and fourth quadrants in system ranking. Even so, the average precision of the best VSM system is roughly twice that of the top TM system, four times the top HITS. The best performing VSM system, measured by average precision, was vlc10 (long query, body text, phrase, and no feedback). The best HITS system was hpm (short host, seed set system of vmc10) for topics 451-500. The best TM system, which differed over topic sets as HITS did, was t221 (top 2 categories, body text, phrase, no feedback) for topics 451-500. The best probabilistic system was pdp0, index from document, short query and no feedback.

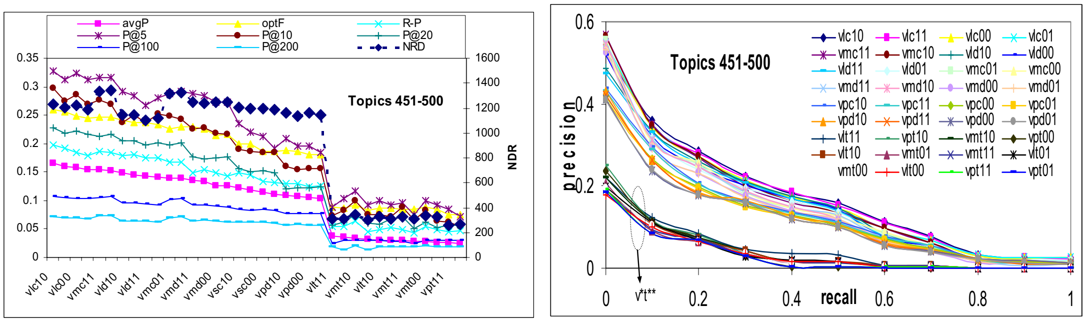

Figure 1. Results of VSM systems for topics 451-500.

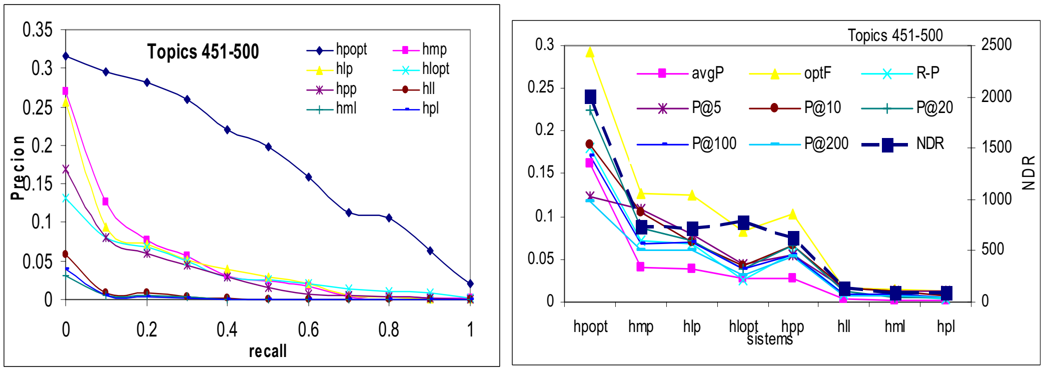

Figure 2. Results of HITS systems for topics 451-500. (hpopt, is produced through all relevant document previous known).

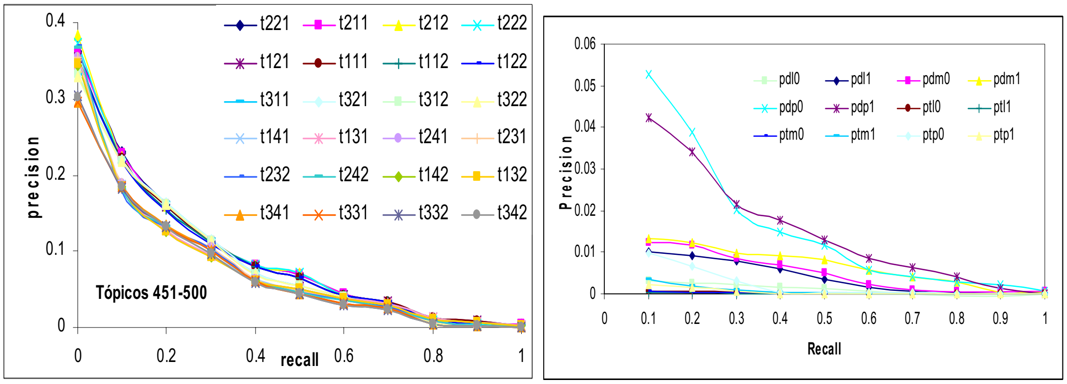

Figure 3. Results of TM (left) and Probabilistic (right) systems for topics 451-500.

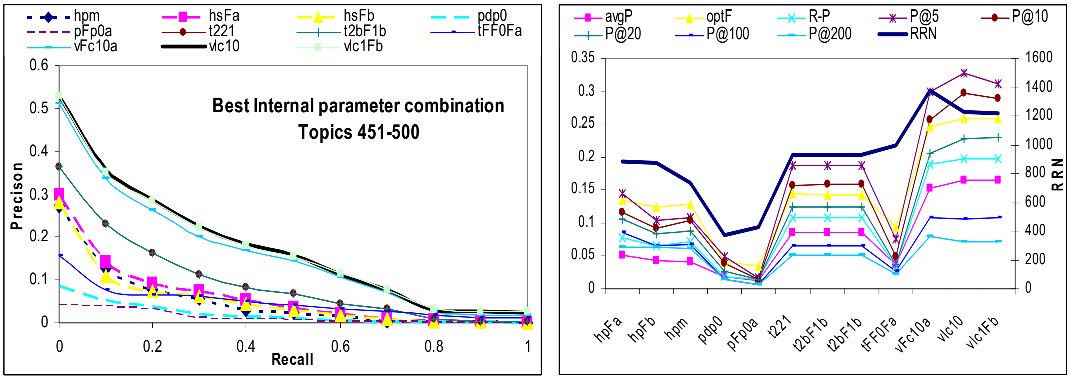

Figure 4. Resume internal combination of parameters for topics 451-500.

Notation for all systems is in tables 1 to 4; RRN = Total Number of Relevant documents; avgP = average precision averaged over queries; optF = optimum F; R-P = R-Precision; P@*k* = Precision at rank *k.*

In general, the most influential system parameter appears to be the query length. It is interesting to note that VSM and HITS systems benefit from longer queries, whereas TM systems perform better with shorter queries. Host definition, which determines the elimination of intrahost links and computation of link edge weights, seems to be a crucial parameter for HITS systems.

Internal system parameters combination: We implemented, two of the most common combination formulas, the *Similarity Merge* [10,11,12,13] *Weighted Sum* [14,15,16,17].

The Similarity Merge (SM) combination formula, originally introduced by Fox and Shaw [10,11] and refined by [12, 13], computes the combination score of a document by the sum of normalized component scores boosted by the retrieval overlap. In order to address the issue of combining a large number of systems with uneven distribution across methods, the overlap count was normalized by the number of systems in a given method. Equation (9) below describes the SM combination formula used to merge and rank documents retrieved by different systems: $CS = (\sum NS_i) * \frac{olp}{m(i)}$ (9); where: FS = combination score of a document; $NS_i = (S_i - S_{\min}) / (S_{\max} - S_{\min})$ (10); NS_i = normalized score of a document by system 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 system I; olp = number of systems that retrieved a given document; $m(i)$ = number of systems in a method to which system i belongs. This formulae is identified on figure 4, by suffix 'a' at end.

To compensate for the differences among combination component systems, the *Weighted Rank Sum* (WRS) formula, which uses rank-based scores (e.g. 1/rank) in place of document scores of WS formula, was tested: $CS = \sum(w_i * RS_i)$, (11); where: FS = combination score of a document, w_i = weight of system i, RS_i = rank-based score of a document by system i. This formula is identified on figure 4, by suffix b at end.

Comparing the best performances of SM and WRS combination with the best baseline system reveals some interesting patterns of interplay between the combination formula and the retrieval method. In both VSM and TM combination, WRS closely shadows the baseline system while SM falls below the baseline performance. In HITS combination, however, SM results are the best by all performance measures while WRS seems to overtake and surpass the baseline performance at lower ranks.

4. RESULTS DISCUSSION AND CONCLUSIONS

In order to investigate the effects of various evidence source parameters, 36 text-based systems based on the Vector Space Model, 6 link-based systems using the HITS algorithm, 24 classification-based and 12 probabilistic based systems using Yahoo category term matching approach were implemented to produce 78 sets of retrieval results for each of the 100 WT10g topics. The retrieval results were then combined in a comprehensive manner within each method as well as across methods using a score-based and a rank-based combination formula producing additional 192 VSM, 24 HITS, 120 TM and 72 Probabilistic systems. In total we produce 488 systems using a common platform [2] to avoid big effort on the construction of systems. Analysis of results suggests that *query length and host definition are the most influential system parameters for retrieval performance*.

For VSM and HITS systems that use the VSM results as the seed documents, longer queries produced far better results than shorter queries, while shorter queries affected better results in TM systems. The host definition, which directly influences both the elimination of intrahost links and link weight computation of the HITS algorithm, turned out to be a crucial parameter for HITS systems, with the shorter definition is clearly superior to the longer definition.

For HITS systems, the quality of the seed document set, both in the number of relevant documents and the richness of link topology appeared to be vital for their effectiveness. Even the optimum HITS system, using the seed set of all known relevant documents, produced disappointing results due to many queries that produced only a small number of relevant documents and the possibly truncated and spurious link topology of WT10g. In fact, 83 out of 100 seed sets produced by the best VSM system were composed of 83% or more non-relevant documents, which severely handicapped the maximum performance threshold of HITS systems. Among the retrieval systems tested, VSM systems clearly outperformed other systems, with TM systems showing better results than HITS systems. In general, average precisions of VSM systems were roughly twice as good as TM systems and four times the average precisions of HITS systems. Internal combination of VSM and TM systems behaved similarly in that combination detracted from the baseline performance although combining TM system results degraded baseline results much more severely than VSM combination when using the SM formula. Combinations in general show more relevant documents identified and SM formula shows improvements at HITS systems and degradation of results at TM and VSM systems. WRS formula shows similar results in all systems (figure 4).

REFERENCES

[1] http://www.ted.cmis.csiro.au/TRECWeb/ access_to_data.html.

[2] Ferreira, João; Silva, Alberto; Delgado, José (2004). *Infraestrutura modular de teste para pesquisa de informação*. Proceedings of the IADIS Conferencia Ibero-Americana WWW/Internet 2004 - October 7 - 8, 2004.

[3] Yang Y. e Pederson J. O. (1997). *Feature selection in statistical learning of text categorization*. Proceedings of the 14th International Conference on Machine Learning.

[4] Sparck J. K. (1971). Automatic Keyword Classification for Information Retrieval. London: Butterworth.

[5] Fishburn P. C.. , 1970 *Utility theory for decision making*, John Wiley, New York.

[6] Wong S. K. M. Yao Y. Y. Salton G. & Buckley C. , 1991. *Evaluation of an adaptive linear model*. JASIS, 42 723-730.

[7] Sumner, R. G., Jr., & Shaw, W. M., Jr., 1997. An investigation of relevance feedback using adaptive linear and probabilistic models. In E. M. Voorhees & D. K. Harman (Eds.), The Fifth Text REtrieval Conference (TREC-5).

[8] Sumner, R. G., Jr., Yang, K., Akers, R., & Shaw, W. M., 1998, Jr.. *Interactive retrieval using IRIS: TREC-6 experiments*. In E. M. Voorhees & D. K. Harman (Eds.), The Sixth Text REtrieval Conference (TREC-6).

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

[10] Fox E. A. & Shaw J. A. , 1994. *Combination of multiple searches*. In D. K. Harman (Ed.). TREC-2.

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

[12] 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.

[13] Lee J. H. , 1997. *Analyses of multiple evidence combination*. Proceedings of the ACM SIGIR Conference on Research and Development in IR, 267-276.

[14] Bartell, B. T., Cottrell, G. W., & Belew, R. K. , 1994. *Automatic combination of multiple ranked retrieval systems*. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.

[15] Larkey, L. & Croft, W. B. , 1996. *Combining Classifiers in Text Categorization*. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 289-297.

[16] Modha, D. & Spangler, W. S., 2000. *Clustering hypertext with applications to Web searching*. Proceedings of the 11th ACM Hypertext Conference,143-152.

[17] Thompson. P., 1990. A combination of expert opinion approach to probabilistic information retrieval, part 1: The conceptual model. Information Processing & Management, 26(3), 371-382.