A MODULAR PLATFORM APPLICABLE TO ALL STATISTICAL RETRIEVAL MODELS

J.Ferreira[1] , A. Silva[2] and J.Delgado[3]

[1]Department of Mathematics, ISEL, Lisbon, Portugal

[1]jferreira@deetc.isel.ipl.pt

[2] [3]Department of informatics, IST, Lisbon, Portugal

[2]alberto.silva@acm.org

[3]jose.delgado@tagus.ist.utl.pt

ABSTRACT

This paper proposes a standard testing platform for Information Retrieval (IR) that can be used for testing different IR statistical models in a controlled environment or in the Web. The development of a standard system avoids the effort of developing a specific testing system to validate each method or model on the IR field of activity, working as a common platform. Examples of applications on filtering, classification and retrieval of information are presented.

KEYWORDS

Framework IR models, Retrieval, Filtering, Classification

1. INTRODUCTION

How to find information on the Web? This is an old issue far from being solved. Several methods and algorithms have been tested without achieving completely satisfactory results, [1,2]. Each time a new method or algorithm is created, a big effort is applied in the construction of a retrieval system to test that method or algorithm. There are no standards of retrieval systems and just a few open source platforms. One initiative regarding a common platform was the SMART project (1960), for testing the vectorial retrieval models, Okapi for probabilistic models and Inquery (1994) for inference network models. Recently, two important frameworks appeared: Lemur (2000), built for testing language models, and Terrier (2001) built to implement DFR (Divergence for Randomness Framework). The discussion in this paper is focused on the necessity of defining an index scheme which can cover all statistical IR models and propose a generic IR testing system platform independent of each IR group. This work can be the starting point to develop standard retrieval systems that can be available in a distributed manner, controlled by organisms like TREC. The main advantages of standardizing modular testing retrieval systems are: (1) less work for testing methods and algorithms; (2) a common infrastructure providing better evaluation of results.

This modular platform is described in the following sections: (2) Platform concept; (3) Application of platform; (4) Results in a controlled environment; (5) Conclusions.

2. PLATFORM CONCEPT

The system contains a number of modules, which can be organized around 3 layers, developed in a distributed “plug-in” architecture. The first layer is responsible for creating information representatives and user interaction; the second is the layer of IR models; the third layer is

application oriented. An appropriate communication space allows communication between different layers.

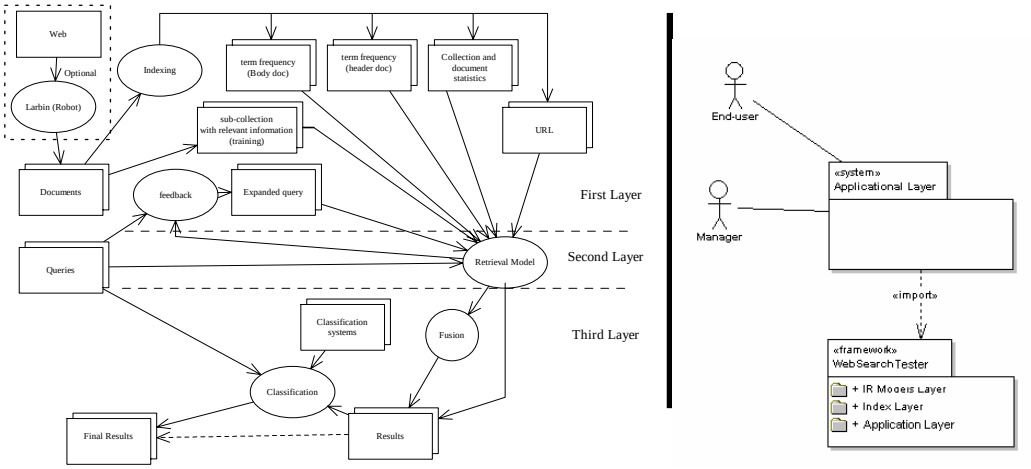

Figure 1: Main components of WebSeachTester.

Our system was built in a modular way, using 3 levels of modules (see figures 1 and 2).

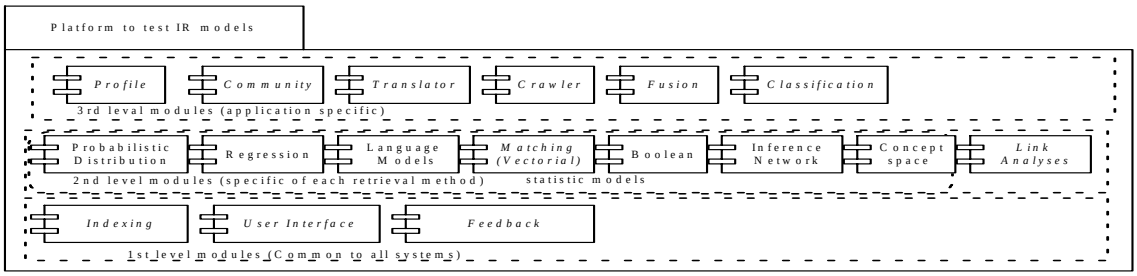

Figure 2: Main modules of the Platform to search Information over the Internet.

2.1 Nuclear Modules – first layer

Indexing: The main processes are described in Figure 3:

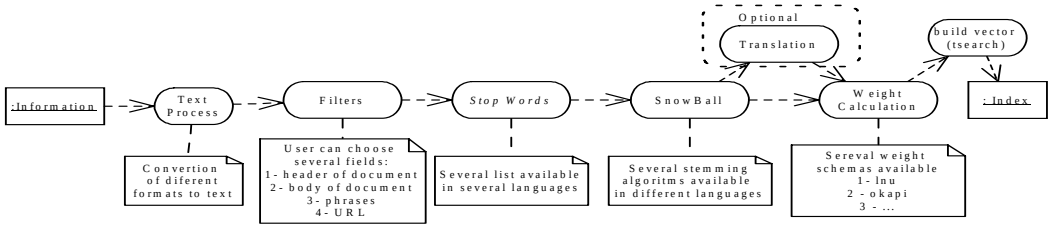

Figure 3: Description of the main steps of the indexing process.

Text process: all documents are converted to text through appropriate converters (e.g., pdf2text, html2text);

Filters: identify specific information such as headers (frequency of these terms are multiplied by 10), URL, phrases based on the electronic dictionary of Roger Mitton, Oxford Advanced Learner http://www.oup.com/elt/global/products/oald/

Stop words are removed, based on the list of the 390 not relevant terms released by WAIS (*Wide Area Information System*) <http://www.ai.mit.edu/extra/the-net/wais.html>.

Snowball <snowball.tartarus.org>: stemming program in which we implemented Porter's algorithm. There are different language stemmers available.

Translation, performed by wordtrans <wordtrans.sourceforce.net>. These modules were introduced so that the system can obtain cross-language results. It is a 3rd layer module applied at indexing time.

Weight calculation: from term frequency. Several approaches can be implemented (e.g. okapi, lnu). Statistical information about documents and collections is stored in a database.

The Indexing module was created based on: (1) OpenFTS <openfts.sourceforge.net>, a front-end that integrates snowball, stop words removal, phrase construction through dictionary and user interface; (2) tsearch2 <http://www.sai.msu.su/~megera/postgres/gist/tsearch/V2> have pre-defined filters, interface to a data base through tsearch vector; (3) weight term calculation: several approaches based on term frequency were easily implemented; (4) data base postgresql. The weight calculation module is built, and the others modules are common and just need to be integrated. All these choices are based on flexibility and integration rather than on speed and performance.

Feedback: The 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 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 [3], is to find a solution vector that will rank a more-preferred document before a less-preferred one [4]. The solution vector is obtained via an error-correction procedure, which begins with a starting vector $q_{(0)}$ and repeats the cycle of "error-correction" until a vector that ranks documents according to the preference order estimation based on relevant feedback is found (Wong et al., 1991). The error-correction cycle is defined by: $\mathbf{q}_{(i+1)} = \mathbf{q}_{(i)} + \alpha\mathbf{b}$, (1); where α is a constant, and *b* is the difference vector, resulting from subtracting a less-preferred document vector from a more-preferred one [5]. The choices for the constant α and the starting vector $q_{(0)}$ are taken from various experiences on TREC [5,6].

2.2 Modules - Second layer

These modules are related to each specific IR model based on statistical properties of documents: Vectorial (lnu measure); probabilistic distribution; Boolean; Fuzzy model; LSI; Regression, probability based on document generation (Okapi measures implemented); probability based on query generation, concept space and inference network model. For more details see <http://www.deetc.isel.ipl.pt/matematica/jf/ir.htm>

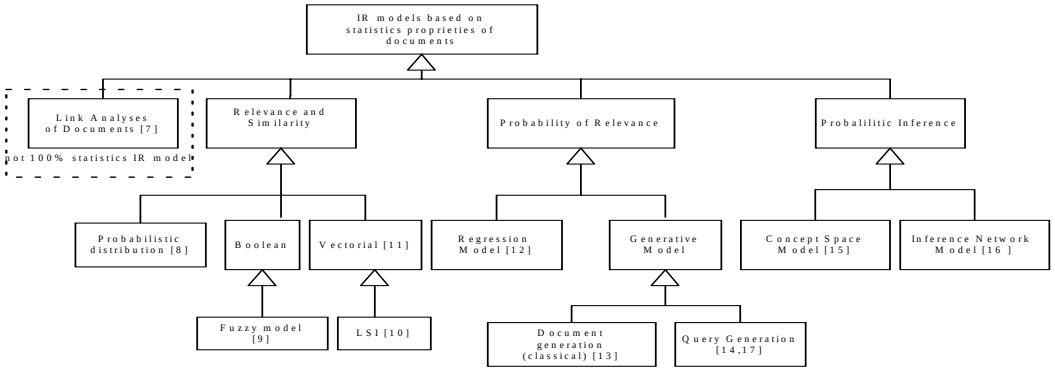

Figure 4: Main IR models based on statistical proprieties of document.

Table 1: List of symbols used in the formulas.

$\overline{dl}$ – is the average lenght of doc. in the collection; n-nº terms in the query; D total number of doc. in the collection; dl_i-length of doc. i; $f_t = \sum_{i=1}^{D} f_{ti}$ – nº of occurrences of term t in the collection; f_{ti} – nº of occurrences of term t in the doc.i; f_{qt} – nº of occurrences of term t in the query; t_n – nº of doc. in which term t occurs; N-nº of terms in the collection; M- nº of terms in common between query and doc.; $idf_t = \frac{D - t_n}{t_n}$; $\lambda_t = \frac{f_t}{N}$; $\|d\|$ – nº of terms of doc.

Matching (Vectorial model) [11]

The tsearch function allows the extraction of values from a data base and the match of the document and the query representatives through Lnu document term weight (1):

$$w_{ti} = \frac{(1+\log(f_{ti}))\Big/(1+\log(\frac{dl_i}{n_i}))}{(1.0-slope)*pivot+slope*t_n}\ (2);\ w_{qt} = \frac{(\log(f_{qt})+1)*idf_t}{\sqrt{\sum_{j=1}^{n}\left[(\log(f_j)+1)*idf_j\right]^2}}\ (3);\ RSV(d_i,q) = \sum_{t=1}^{n} w_{ti} * w_{qt}\ (4)$$ with the

slope of 0.3 for document terms [20]. Documents and profiles were clustered by their inner product (4). In profile clustering, the weight of the document is replaced by transpose of profile.

HITS

The connections model identifies firstly the seed groups using the vector model. The hub and the authority measurements of these groups are calculated through the HITS algorithm [3]. The address definition was based on the document's URL, cutting it in the first occurrence of the slash ("/") (short address format) and the long format until the last occurrence of the slash ("/").

The main system parameters are the choice of the method to create the seed group, the address definition and the way hubs and authorities are measured. The total number of possible systems is 2x3=6.

$$a(p) = \sum_{q \to p} h(q) \times auth\_wt(q,p)\quad (5);\ h(p) = \sum_{p \to q} a(q) \times hub\_wt(p,q)\quad (6)$$ *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.

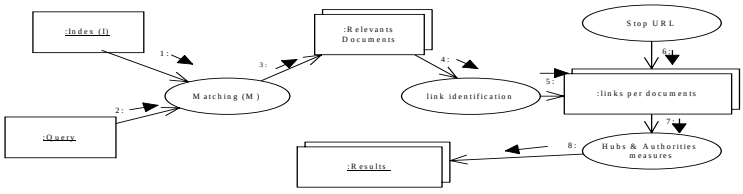

Figure 6: Main modules of link analysis system.

Probabilistic Classical document generation (Okapi formulae) [13]

$$w_{ti} = \frac{(k_1+1)\,f_{ti}}{k_1\left((1-b)+b\frac{dl_i}{\overline{dl}}\right)};\ w_{qt} = \frac{(k_3+1)\,f_{qt}}{k_3+f_{qt}}\log_2\left(\frac{D-t_n+0.5}{t_n+0.5}\right);\ RSV(d_i,q) = \sum_{t=1}^{n} w_{ti} * w_{qt}\quad (7)$$

$k_1 \in [1,2]; b \approx 0.75; k_3 \in [0,1000]$ $\overline{dl}$ – is the average size of doc. i the collection; dl doc. size (bytes); N total number of doc. in the collection;

Query generation, language model approach [14,17]

$\log p(q \mid d) = \sum_{\substack{t_i \in d \\ t_i \in q}} [\log \frac{p_{seen}(t_i \mid d)}{\alpha_d \; p(t_i \mid C)}] + n \log \alpha_d + \sum_i \log p(t_i \mid C)$ (8); $p_{seen}(t_i \mid d)$ weight t_i; $p(t_i \mid C)$ weight of idf_t; $n \log \alpha_d$ doc. normalization; $\sum_i \log p(t_i \mid C)$ ignored (constant for each query)

P(w|d) is estimated based on 3 approaches:

(1) LM1: Jelinek-Mercer method: Shrink uniformly toward p(w|C)

$p(t_i \mid d) = (1-\lambda)\, p_{ml}(t_i \mid d) + \lambda \; p(t_i \mid C)$ (11) ; $p_{ml}(t_i \mid d) = \frac{f_t}{|d|}$ (9)

(2) LM2: Dirichlet prior (Bayesian): Assumes pseudo counts μp(t|C)

$p(t_i \mid d) = \frac{f_t + \mu\, p(t_i|C)}{|d| + \mu} = \frac{|d|}{|d| + \mu}\, p_{ml}(t_i \mid d) + \frac{\mu}{|d| + \mu}\, p(t_i \mid C)$ (10) ($\mu = 1600$ tuned for WT10g collection)

(3) LM3: Absolute discounting: Subtract a constant δ: $p(t_i \mid d) = \frac{\max(f_t - \delta, 0) + \delta |d|_u \, p(t_i|C)}{|d|}$ (11)

Logistic Regression [12]

Builds a regression model for relevance prediction based on a set of training data retrieval. The probability estimation is obtained by:

$$\log \frac{P(R \mid q, d)}{P(\overline{R} \mid q, d)} = \alpha + \sum_{i=1}^{6} \beta_i \times x_i \quad (12)$$

$X_1 = \frac{1}{M} \sum_{k=1}^{M} \left(\log f_{qk} \right)$; $X_2 = \sqrt{n}$; $X_3 = \frac{1}{M} \sum_{k=1}^{M} \log(\mathrm{f_{ki}})$; $X_4 = \sqrt{dl}$; $X_5 = \frac{1}{M} \sum_{k=1}^{M} \log idf_k$; $X_6 = \log M$ X_1- Average Absolute Query Frequency; X_2-Query Length; X_3-Average Absolute Document Frequency; X_4-Document Length; X_5-Average Inverse Document Frequency; X_6- Number of Terms in common between query and document – logged. α - intercept term of the regression

Inference [16]

The standard probabilistic model assumes that you can not estimate P(R|D,Q). Instead, it assumes independency and uses P(D|R). Bayesian network can estimate P(R|D,Q), and intends to capture all the significant probabilistic dependencies among the variables represented by nodes in the query and document networks. Given the priors associated with the documents, and the conditional probabilities associated to internal nodes, we can compute the probability (belief) associated with each node in the network

$$w_{ti} = 0.4 + 0.6 * \frac{f_{ti}}{f_{ti} + 0.5 + 1.5 \frac{dl_i(d)}{\overline{dl}}} \frac{\log\left(\frac{D+1}{t_n}\right)}{\log(D+1)}; RSV(d_i, q) = \sum_{t=1}^{n} w_{ti} * w_{qt} \quad (13)$$

2.3 Third Layer Modules

Classification

To leverage the classification of information, the Web crawler Larbin (3rd level module) was customized to spider the Yahoo site in the fall of 2002. In addition to reproducing a cleaned-up version of Yahoo locally, the crawler created "sitemap" files in each of the major Yahoo categories to encapsulate the classification of information. How to leverage the information captured in the sitemap files, which essentially includes the classification regarding hierarchy, category labels, site titles and site descriptions, is a research question in its own right. The Term Match (TM) was implemented based on matching of terms query-category. The first step of the TM method 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.

Fusion

On fusion, we have implemented two new approaches based on the most common fusion formulas with small changes, the Similarity Merge (SM) $P_1(R \mid D,Q)$ and Weighted Sum (WS) $P_2(R \mid D,Q)$ [18,19]. We also propose a mixed formula to use both approaches.

$$P_{\{1,2\}}(R \mid D,Q) \propto \sum_{i=1}^{n} \sum_{j=1}^{2^n-1} \left(\left\{ \frac{s_i - s_{mim}}{s_{\max} - s_{mim}}, \frac{1}{rank_i} \right\} * w_{ij} \right) \quad (14)$$

$w_{ij} = avg(P_{ij}@10) * olp_j$; olp_j-overlap at partition j

$avg(P_{ij}@10)$ average precision at top 10 doc.

n number of IR systems to combine; $2^n - 1$ number of partitions possible with n system less null set

(Ex. A,B and C systems: we have AB, AC,BC,ABC,A,B,C partitions $2^3 - 1 = 7$).

Profile

The user profile represents its stable information needs, and can be used on: (1) filtering systems; (2) retrieval information (documents are ranked against the user profile); (3) information access.

The profile is built in the same way as a query, but since it is stable we can profit from the user feedback models. The profile can be established from: (1) introduction of terms from the user; (2) choice of communities that the user considers to match his own interests; (3) terms can be chosen from a classified space. The user can also establish a negative profile that reflects topics on which he isn't interested at all.

Communities (Groups of similar profiles)

Communities can be applied in several fields: profile definition, collecting relevant information for authors and distributors of information, dissemination of information through previously identified communities. The vector comparison system does the comparison between different profiles. The similar profiles will be treated with the distance function and then evaluated, based on the experience and the specificity of the topic. To become effective, the communities have to be analyzed and confirmed by a human authority, due to the high complexity involved in the process. Two main services are available: the user receives information about the arrival of a relevant document according to his profile; when a new profile appears, all the users from the communities to witch this new profile can belong are informed. For more information see [21].

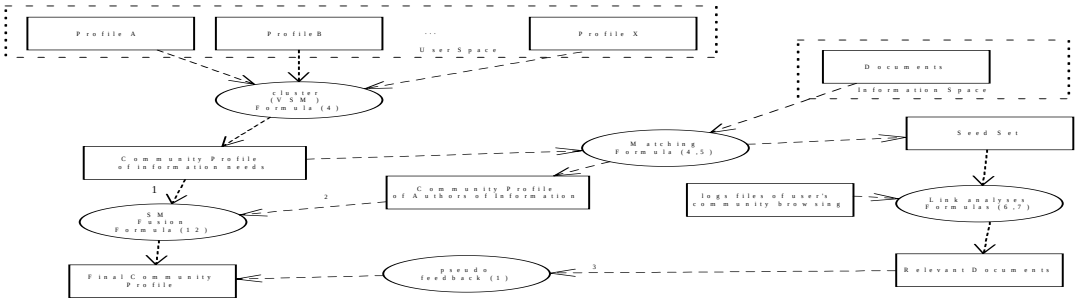

Figure 7: Process of Communities Identification.

Translator

We integrated a translator, Wordtrans <wordtrans.sourceforge.net>, that provides the translation of the questions in spite of the usual limitations of a translator (text association is not done).

Five different languages are available: German, French, Italian, Spanish and Portuguese. If one word is not found in the dictionary, it appears in its original form.

Robot

Was integrated Larbin, <larbin.sourceforge.net>. Target source and distance of hyperlink follow-up is configured. User feedback is incorporated, with identification of target information sources.

3 Aplications of the platform

3.1 Filtering Systems

Filtering systems send information periodically to the registered users according to their stable interests. Should have as inputs: (1) user profiles; (2) documents; (3) user communities. The filtering process is based on matching the user's profile with the documents' representatives using the vector model. The result is a group of documents identified by a positive profile without the documents identified by a negative profile. If a document is simultaneously identified by positive and negative profile, the result with bigger measure remains. The user feedback model is used to change and tune user profiles.

Filtering systems detects the following events: (1) new relevant documents; (2) changes identified in old relevant documents; (3) new users on communities; (4) change on user profile.

MyNewsPaper; Personal NewsPaper

This system searches news based on the user profile and sends them to the user regularly. The user can define the periodicity, the mail, the positive profile (topics that the user is interested on) and the negative profile (topics that the user is definitely not interested in). From the platform we use the following modules: Indexing; user interface; matching; profile, robot, communities and user feedback (used to refine user profile). Larbin was configured to pick up daily information from the Portuguese newspaper "O Público" <www.publico.pt>, (more information sources could be added). The Community module based on matching of user profile identifies user communities which can be used for diffusion information systems and it is valuable information for authors and editors of information. At this moment we are changing the user interface to put the system on line.

MyTV: Personalized TV programs adviser

MyTv is a personalized TV programs notification service of Portuguese TvCabo <www.tvcabo.pt>. The profile is defined using the list of available programs as a basis, or using terms that make part of a previous defined list. The main challenges of this system are the identification of subjects in generic channels and the retrieval in different languages (Portuguese, English). MyTV uses modules: indexing; matching; user interface, classification; translation; profile; communities and user feedback

MyEnterprise News (find enterprise news through Internet)

The aim was to develop a program that is able to identify, amongst all the news, information that is relevant for a certain company, its field of activity and its competitors. These data can be used on taking strategic decisions. The degree of complexity of the described exercise is really high given the availability of relevant information in several languages from many sources. In this specific exercise six languages have been considered: English, Portuguese, Spanish, Italian, German and French. The automatic translation within these six languages is possible.

3.2 Classification systems

MyDocument

The system manages the documents of a department. The inputs are the documents and the classification system. The classification process is performed using the vector comparison of the terms available in the classification system with the titles of the documents. In the case that no

relevant documents are found, the classification has to be done manually. This system uses the following modules: Indexing; user interface; matching; classification; user feedback; robot.

3.3 Retrieval systems

Retrieval systems can be used in a controlled environment (using defined queries where results are previously known using a certain collection), to measure system output. This subject is analyzed in the next section. Other possibility is to perform a search in the Web but the results are difficult to measure.

4 Results of Retrieval Systems in a controlled environment

In a controlled environment (using TREC WT10g collection), we constructed several retrieval systems and studied the behavior of systems' parameters and fusion of results. For detailed results see [22].

Vector system

This system was build from the modules: indexing; matching; user interface and user feedback. Main systems parameters are: (1) query length (long(l), medium(m) and small(s)), (2) indexing information from headers(t), body document (c) or complete document (d), (3) phrases (yes or no); (4) user feedback(1(yes)/0(no)). The combination of all parameters originates 36 different systems. The query length is the most important parameter. Indexing headers produces bad results because often headers are intentionally misleading or constructed with less attention to content and more attention to appearance than purely textual documents. For example, Web page titles are often missing, texts not relevant to the central document theme become headings for cosmetic, navigational, or commercial reasons, and meta tags, when present, are sometimes stuffed with spam words to promote the documents with misleading information.

Link-based Retrieval Results

This system was build from the modules: indexing; matching; user interface, HITS and user feedback. The main system's parameters are: (1) host definition (l-long; s-small) (2) seed set (vectorial, (l-long; m-medium; s-small) query, body text, phrase, no feedback. The combination of all parameters originates 6 different systems. Recall-precision graphs clearly show the influence of host definition on retrieval performance for HITS systems. The shortest host definition is obviously far superior to longer definitions (over 10 times better in average precision). The overall performance level of HITS systems is somewhat disappointing. HITS systems with short host definition achieve only one fourth the average precision of the seed VSM systems. This indicates that probably the link structure of WT10g collection is incomplete.

Probabilistic (Classical, Okapi formulae)

This system was build from the modules: indexing; matching; user interface, Probabilistic and user feedback. Main systems parameters are: (1) query length (s/m/l); WT10g index (1 –header with phrase; 2-document with phrase; 3 –header without phrase; 4-document without phrase); feedback (1(yes)/0(no)). Combination of all parameters gives 12 different systems. Results were similar to those of the vector model.

Language Model, Inference, Logistic Regression

In both systems we tested different query lengths (short, medium and long) and indexing from full documents. Parameters' tuning is crucial for system performance and depends on the collection used.

Fusion of retrieval systems

Internal-Method fusion is a combination of systems with the same IR model. External-Method fusion combines results across models. The differences in retrieval methods that generate different retrieval outcomes seem to be influenced both by internal and external fusion, where the system results were combined within and across retrieval methods. Interestingly, the only internal fusion that enhanced the baseline performance of the best fusion component results occurred with the

worst performing HITS systems. Internal fusion of VSM, Probabilistic and Classification systems behaved similarly in that fusion detracted from the baseline performance, although combining Probabilistic and Classification system results degraded baseline results much more severely than VSM fusion when using the SM formula. In HITS fusion, the score-based SM formula produced better fusion results than those of the rank-based WS formula, which was opposite with respect to the other systems. External fusion combinations produced results in between upper and lower threshold performance levels determined by the baseline systems of the methods combined.

The different outcomes of SM and WS fusion formulas, which were observed in internal fusion, appeared in external fusion results as well, although the SM formula results seemed to be more stable across methods than WS formula results. In general, the WS formula appears to have an advantage over the SM formula, which worked better with HITS systems. Since the optimization of the fusion formula was not the main focus of our work, an in-depth investigation into the different behaviors of fusion formulas was not conducted. Instead, we considered as potential causes for the different outcomes the main differences between SM and WS formulas, which were the SM's tendency to differentiate between documents in rank proximity, SM's heavier emphasis on the overlap count, and the WS weighting of fusion component contributions based on past performance.

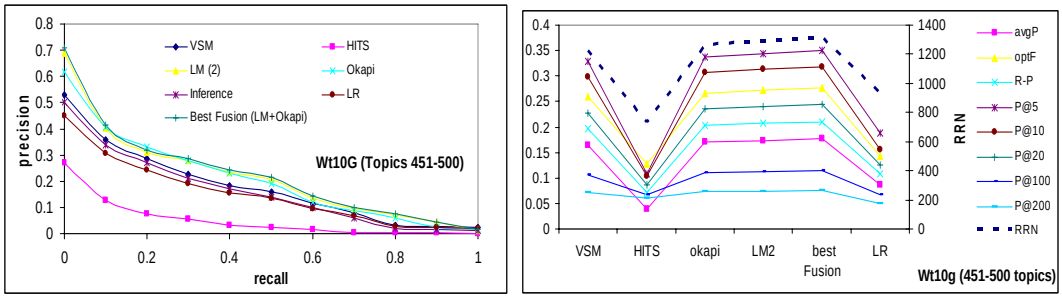

Figure 8: Best results of main IR models implemented. RRN - total number of relevant documents retrieved.

5 Conclusions

A modular infrastructure that allows the testing of Web research systems and methods was described, as well as the conceptual discussion of different researches, filtering and classification of information systems and their possible implementation methods. This platform allows the development of testing systems with a lower effort using and integrating existing modules. The comparison of the results is also more accurate once the modules used to test different systems are the same.

We have also shown how to implement, in the same platform, systems based in the vector models, in connection models and in classification models, and additionally the possibility of combining results trying to identify the most relevant parameters in the research.

6 References

[1] Korfhage Robert R.. *Information Storage and Retrieval*. John Wiley e Sons Inc., 1997.

[2] Yates R. B. E Neto B. R.. *Modern Information Retrieval***.** Addison-Wesley Pub Co., 1999.

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

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

[5] Sumner, R. G., Jr., & Shaw, W. M., Jr.. *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)*, 1997.

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

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

[8] Wong, S. K. M. and Yao, Y. Y. (1989). *A probability distribution model for information* retrieval. *Information Processing and Management*, 25(1):39–53.

[9] Bookstein A. (1985). *Probability and fuzzy-set applications to information retrieval.* Annual Review of Information Science and Technology 20 117-151.

[10] Dumais S. T. (1994). Latent Semantic Indexing (LSI) and TREC-2. In D. K. Harman (Ed.) Proceedings of the 2nd Text REtrieval Conference (TREC-2) 105-115.

[11] Salton G. (1971). The SMART Retrieval System - Experments in Automatic Document Processing. Englewood Cliffs NJ: Prentice-Hall Inc.

[12] Fox, E. (1983). Expending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. PhD thesis, Cornell University.

[13] K. Sparck-Jones, S. Walker and S.E. Robertson (2000), A probabilistic model of information retrieval: development and comparative experiments. Information Processing &Management 36(6), pp. 779-840, 2000

[15] Wong, S. K. M. and Yao, Y. Y. (1995). On modeling information retrieval with probabilistic inference. ACM Transactions on Information Systems, 13(1):69–99.

[14] Ponte, J. and Croft, W. B. (1998). A language modeling approach to information retrieval. In Proceedings of the ACM SIGIR'98, pages 275–281.

[16] Turtle, H. and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems, 9(3):187–222.

[17] Zhai, C. and Lafferty, J. (2001). Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management(CIKM 2001)*, pag 403–410.

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

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

[20] Buckley C. Singhal A. e Mitra M. (1997). *Using query zoning and correlation within SMART: TREC 5*. In E. M. Voorhees & D. K. Harman (Eds

[21] Ferreira J., Silva A.and Delgado J.. *Fusion methods to find Web Communities*, In Proceedings of the Web based Communities 2005, 23-25 of February 2005, Alvarge (Portugal), www.iadis.org/wbc2005/.

[22] Ferreira, J.. *Retrieval Information on the Web*. PhD Thesis, IST, Portugal (in Portuguese), September 2004, IST.