Mining Information Retrieval Results

Significant IR parameters

Jonathan Compaoré
Adji Maïram Gueye
Institut National des Sciences Appliquées
Université de Toulouse
Toulouse, France
{jcompaor/amgueye}@etud.insa-toulouse.fr

Sébastien Déjean
Institut de Mathématique de Toulouse
Université de Toulouse
Toulouse, France
sebastien.dejean@math.univ-toulouse.fr

Josiane Mothe
Joelson Randriamparany
Institut de Recherche en Informatique de Toulouse
Université de Toulouse
Toulouse, France
{mothe/randriamparany}@irit.fr

***Abstract*—This paper presents the results of mining a large set of information retrieval results. The objective of this study is to determine which parameters significantly affect search engine performance. We focus on the main features of information retrieval: indexing parameters and search models. Statistical analysis identifies the retrieval model as the most important parameter to be tuned to improve the performance of an information retrieval system. We also show that the significant parameters depend on the difficulty of the topic.**

Keywords-information retrieval; data mining; parameter analysis; performance prediction.

I. Introduction

Information retrieval systems are generally evaluated considering their effectiveness. While evaluated in international campaigns such as TREC [1] and CLEF [2], etc., consider the Cranfield evaluation model [3]. This model consists of a collection of documents on which a query is evaluated, a set of queries and the list of relevant documents for each of these queries. Evaluation frameworks also imply performance measurements that will be helpful to compare systems.

Evaluation campaigns have contributed a lot to the Information Retrieval (IR) field [4][5]. They clearly contribute to the definition of new models and new processes such as blind relevance feedback for example [6]. From the published performance results it is possible to know which systems perform best on the set of topics suggested by the evaluation program. In addition, from the system description published onin the associated papers, it is generally possible to know in detail the type of systems used and even the various parameters used in a specific experiment.

On the other hand, what the results hide is the individual contribution of a given component. Indeed from these experiments, it is not possible to know what the impact of the indexing is, nor a given parameter. The impact of parameters in models is evaluated when defining the model. For example, Zhai and Lafferty [7] present the precision when various parameters of smoothing methods for language modeling vary. In this paper our goal is different, we aim at discovering if some parameters are significantly correlated with some performance measures. To discover such information, we use a platform that implements various indexing schemes and search models and that allow one to modify parameter values. Then we process the same topics using these various system configurations and evaluate the results in terms of effectiveness. We then analyze the resulting data with the aim of finding associations between system parameters and performance measures.

The rest the paper is organized as follows. In Section 2, we present related works. In Section 3, we describe the platform we use and the way we obtain the data to be analyzed. In Section 4, we present the preliminary results we obtained using one performance measure MAP (Mean Average Precision). Section 5 presents conclusions and outlines directions for future work.

II. Related works

An IR process is composed of various components. A first component is indexing which is done off line, prior to any querying. Indexing aims at deciding which terms will be used to represent document content and to match the document and the query. The indexing unit, stop-word list, stemming algorithm, and term weighting function are among the parameters of indexing. When queried, the IR system has then to match the query with the document. This is done through a similarity function or a ranking function generally based on content similarity (term comparison). Systems vary according to the model used: vector space model [8], Probabilistic model [9][10], LSI [11], language modeling [12]; each model has parameters that can be modified.

Generally speaking, related work analyzes one component or a few components of the retrieval process in terms of its influence on the effectiveness, considering for example MAP.

Kompaoré *et al.* [13], for example, focus on the indexing part. They analyze three types of indexing units applied on French test collections: lemmas, truncated terms and single words. Considering the 284 used in the CLEF French track,

 ISBN: 978-1-61208-162-5

they show that the best results were obtained while considering lemmas. Lifchitz *et al.* [14] analyze the effect of parameters on the LSA model, a model which is based on singular decomposition of the term/document matrix resulting from indexing. They show that normalizations of documents and term frequencies have a negative effect on the results. They also conclude that the optimal truncation (number of dimensions) of the semantic space and the stop word list play a major role.

The "reliable information access" workshop [15][16] focused on variability and analyzed both system and topic variability factors on TREC collections when query expansion is used. Seven systems were used, all using blind relevance feedback. System variability was studied through the different systems by tuning different system parameters and query variability was studied using different query reformulation strategies (different numbers of added terms and documents). Several classes of topic failure were drawn manually, but no indications were given on how to automatically assign a topic to a category.

Bank *et al.* [17] reports various data analysis methods and how they can be used to analyze TREC data. They consider analysis of variance, cluster analyzes, rank correlations, beadplots, multidimensional scaling, and item response analysis. When considering analysis of variance, they consider two effects only: topic and system and one performance measure (average precision). They also used cluster analysis to cluster systems according to MAP they obtained. Even if this preliminary study concluded that none of these methods has yielded any substantial new insights; more recent work [18] has shown that clustering can be used in the case of repeated queries.

III. Generating data

A. TREC data

Since we want to evaluate individual parameters of the search process, it is compulsory to consider the same collection (information needs, documents on which the search is carried out and relevance judgments). International experimental environments, such as TREC, provide such a framework.

The *ad hoc* task was introduced in the earlier years of TREC in 1991. It simulates a traditional IR task for which a user queries the system. The system retrieves a ranked list of documents that answer this query from a static set of documents.

We work in this paper with data used two consecutive years in TREC-7 and TREC-8 collections. The document collection consists of disks 4 and 5 which corresponds to 528 155 documents. A total of 100 topics are used here, topics 351-400 used in TREC-7 and topics 401-450 used in TREC-8. Details on the set of documents and topics can be found in [1].

An example of a TREC topic is presented in Figure 1. In addition to its identifier, a topic is composed of a title, a descriptive and a narrative part. Title only can be used to build a query to be submitted to a system. It is also possible to build the query from other topic part combinations.

<num> 396
<title> sick building syndrome
<desc> Identify documents that discuss sick building syndrome or building-related illnesses
<narr> A relevant document would contain any data that refers to the sick building or building-related illnesses, including illnesses caused by asbestos, air conditioning, pollution controls. Work-related illnesses not caused by the building, such as carpal tunnel syndrome, are not relevant

Figure 1. Example of TREC topic (#396).

B. Terrier platform

Terrier is an information retrieval platform that implements state-of-the-art indexing and retrieval functionalities.

1) Topics

One of the parameters of a search is the topic used. In our experiments, we consider 100 topics, numbered 351 to 450.

From these topics, queries are built using an indexing process (see below). From a topic, three types of queries are built: using title (`T`) only, title and descriptive (`TD`), title, descriptive and narrative (`TDN`), referred as variable *Field*.

2) Indexing

Document indexing is used to extract indexing terms from document contents. Usually, stop-words are removed and remaining terms are stemmed in order to conflate the variant of words into a single form. Indexing terms are weighted in order to reflect their descriptive power.

While indexing documents, they can be cut into several chunks of text. This process has an impact on the term weights. The number of blocs is a parameter (variable *Bloc*). The value is `1` when any document is considered as a unit; the two other values we used are `5` and `10`.

Regarding the weighting schema, it is possible to consider the inverse document frequency. For these reason the parameter *Idf* is set either to 0 (`FALSE`) or to 1 (`TRUE`).

3) Retrieval models

Nine models are implemented in Terrier. We use each of them (variable *Model*): `BB2c1`, `BM25b0.5`, `DFRBM25c1.0`, `IFB2c1.0`, `InexpB2c1.0`, `InexpC2c1.0`, `InL2c1.0`, `PL2c1.0`, `TFIDF`. Details on these models can be obtained at Terrier web site [19].

4) Query expansion

Query reformulation is used in order to improve the initial query so that it can retrieve more relevant documents. Blind relevant feedback is used [6].

We used the three models implemented at the Terrier: `Bo1bfree`, `Bo2bfree`, `KLbfree` (parameter *Ref*).

The number of documents used during query reformulation is a parameter (*DocNb*). It varies in {`0`, `3`, `10`, `50`, `100`, `200`}. The number of documents in which the

terms must occur to be considered as relevant to be used in the expended query is a parameter (*qe_md*). Its value is either `0` or `2`. The number of terms added to the initial query is the last parameter of query expansion; its value is either `0` or `10` (*qe_t*).

Some variables appear to be redundant. For example, *qe_md* and *qe_t* are redundant since as soon as there is an expansion, there will be 10 terms added and they should occurs at least in 2 documents. The variables are presented in Table 1.

TABLE 1. PARAMETERS AND VALUES USED FOR A SEARCH.

Parameters	Meaning	Values
Top	Topic number	351, …, 450
Field	Topic field	`T`; `T+D`; `T+D+N`
Bloc	Size of the indexing bloc	`1`, `5`, `10`
Idf	Inverse document frequency	`FALSE`, `TRUE`
Ref	Query reformulation	`None`, `Bo1bfree`, `Bo2bfree`, `KLbfree`
Model	Retrieval model	`BB2c1`, `BM25b0.5`, `DFRBM25c1.0`, `IFB2c1.0`, `InexpB2c1.0`, `InexpC2c1.0`, `InL2c1.0`, `PL2c1.0`, `TFIDF`
DocNb	Number of documents (reformulation)	`0`, `3`, `10`, `50`, `100`, `200`
qe_md	Minimum number of documents in which the term should appear to be used in the query epension	`0`, `2`
qe_t	Number of terms used in the query epension	`0`, `1`

A combination of these parameters leads to a run that can be evaluated using performance measures.

C. Performance measures

We use the TREC software trec_eval to evaluate each individual run. Measures are computed for each topic. The version 8.1. of trec_eval [20] that we used computes 135 measures. Baccini *et al.* [21] have shown that many performance measures are redundant and that it is possible to keep 6 representative measures that will cover the various aspects of IR evaluation. The remainder of the analysis focus on only one performance measure (MAP) for illustration purpose.

D. Data to analyze

We generate a matrix that is composed of:

- 98650 objects (lines of the matrix). Each line corresponds to one topic processed by a chain of modules (indexing, search) and evaluated according to various performance measures;
- 8 variables (columns of the matrix) that consist in 7 non redundant module parameters (see Table 1 minus *qe_md* and *qe_t*) and 1 performance measure (MAP).

The value in a cell corresponds to the characteristic of the object for the corresponding variable. When it is a system parameter the cell contains the value of the parameter; when it is a performance measure, the cell contains the result of the evaluation.

IV. STATISTICAL ANALYSIS

We aim at identifying which parameters have a significant influence on the performance of the system. To address this question, we first performed an Analysis of Variance (ANOVA) to explain MAP according to each parameter separately. Then using Classification and Regression Trees (CART, [22]) every parameter is jointly analyzed. These methods were applied in three frameworks:

- One global analysis: every topics were considered;
- Two restricted analysis: considering only the easiest (resp. hardest) topics. Easiest (resp. hardest) topics are the ones for which the average AP over the systems is the highest (resp. lowest).

A. Analysis of Variance (ANOVA)

ANOVA tests whether or not the mean of several groups are all equal. In our context, this will result in testing whether the MAP is significantly different when considering various configurations of one parameter.

1) Global analysis

The results of the global analysis are summarized in Figure 2. Parallel boxplots corresponding to the various categories are displayed for each parameter. First, it appears that *Bloc* and *Idf* (white boxplots indicates non significant effect) have no influence on the performance. Considering Field, MAP is lower when using only the title (`T`) of the topic; results are nearly equivalent for `TD` and `TDN`. For *Ref*, the absence of reformulation (`NONE`) seems to be detrimental in relation to one system for query reformulation whatever it is (`Bo1bfree`, `Bo2bfree` or `Klbfree`).

The main comment regarding the retrieval model (variable *Model*) is the bad behavior of `BM25b0.5`, and to a lesser extent of `PL2c1.0`. Then, *DocNb* (the number of documents used in query reformulation) provides the best results with `3` or `10` documents. Then, the MAP decreases as the number of documents used for the reformulation increases.

These results outline phenomenon visible at a global scale considering all the topics. We wanted to see if these results hold for two particular sub-sets of topics: the easiest and the hardest.

2) Analysis of easy and hard topics

Most of the work in IR considers the result globally, averaging the results over a set of queries. On the other hand, some works have shown that system results (e.g. AP) is query dependent [18]. Finally, some studies have focused on hard topics, trying to find ways to handle them better [16]. Finally, Bigot *et al.* [18] show that choosing the best system for individual queries improves results differently according to query difficulty. For that reasons, we decided to consider two types of topics: the hard and easy topics and to analyze the behavior of the parameters according to these

 ISBN: 978-1-61208-162-5

two topic sets.

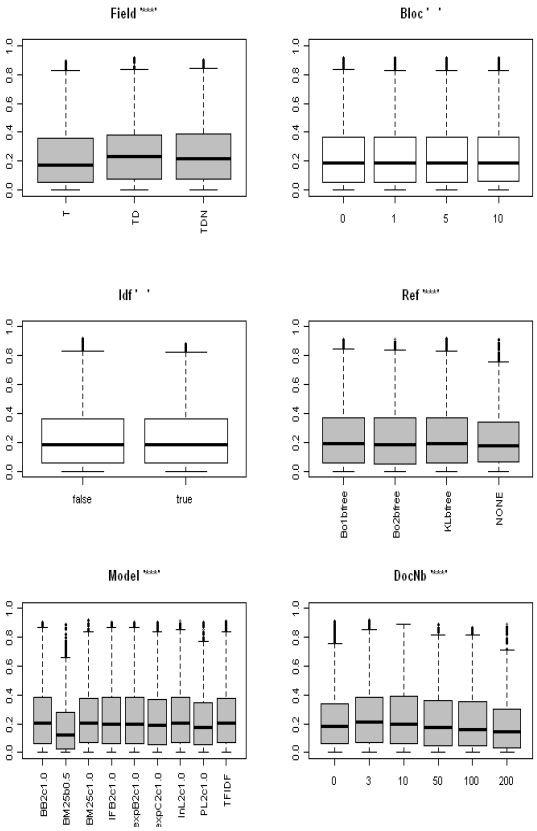

Figure 2. Boxplots representing MAP according to the different levels of each parameter (Field – 3 levels, Bloc – 4 levels, Idf – 2 levels, Ref – 4 levels, Model – 9 levels, DocNb – 6 levels). The symbol near the title indicates the p-value of the test according to the code: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1. Grey boxplots highlights cases where the parameter is highly significant in the ANOVA model (p-value < 0.001).

The "easy" topics are the ones for which the average AP over systems is the highest. Considering the analyzed data, the corresponding AP is higher than 0.45 (considering this value, there is a gap between topics). There are 13 topics in the easy topic set. In the same way, hard topics correspond to topics for which the mean of AP over systems is the lowest. AP is lower than 0.045. There are 19 topics in the hard topic set.

The results of the restricted analysis highlights more heterogeneity in the behavior of MAP according to parameters. For the easy topics, 4 of the 6 parameters have a statistically significant influence on the MAP.

The most visible phenomenon is the very particular profile of the *Model* parameter: using `BM25b0.5` significantly deteriorates MAP. This is also the case for the hard topics but it also appears that some retrieval models, such as `ExpC2c1.0` or `BB2c1.0` are more appropriate to improve MAP.

The influence of the field used when indexing topics (`T`, `TD` or `TDN`) seems to have more impact for hard topics. The dispersion of the MAP values is also higher when considering hard topics. Indeed generally speaking, when considering hard topics there is a larger range of values than when considering easy topics. That is to say, there is no failure when considering easy topics whereas there are some good results for hard topics.

Interesting enough, the impact of the number of documents when using query reformulation is not the same over the two restricted analysis: while `3` or `10` documents seems to be the best choice for easy topics, `0` to `3` is more appropriate for hard topics.

Hard topics also exhibit two other configurations to use to improve MAP: the query reformulation proposed by `Bo1bfree` and the *Idf* set to `FALSE`. However, let's note that even if the MAP is increased it remains relatively low.

B. Classification and Regression Trees

In the previous section, we analyzed the influence of each parameter of the IR process on the results. In this section, we try to sketch a strategy on parameter tuning during the IR process in order to maximize the performances (according to MAP) of a search. To address this question, we also want to deal with the parameters simultaneously and to consider potential interactions between them. This purpose can be achieved by using CART [22]. The main idea in CART lies in the construction of a decision tree by splitting successively the observations depending on the values of the dependent variables which is numeric for regression and categorical for classification. Details can be found in the original article by Breiman *et al.* [22] or in a more recent review [23]. We opted for the classification version of CART in order to identify the most important parameters to be tuned to obtain better results without quantifying the resulting value. Implementation was performed using the *rpart* package [24] of the R software [25].

1) Data

Considering CART in the classification framework required MAP values to be converted into qualitative information. We used the quartiles to divide the range of the MAP into 3 classes. Then, we code the values according to which interval they fall in. We use three tags: "*Bad*" (MAP lower than the first quartile), "*Average*" (MAP between the first and the third quartile) and "*Good*" (MAP greater than the third quartile). Table 2 reports the values of MAP corresponding to these tags according to the type of topics used. Indeed, we considered three sets of queries, like in subsection IV A. Global data consists of the all set of topics, easiest topics and hardest topics corresponding to the sets defined in Section 4.

TABLE 2. MAP VALUES DEPENDING ON THE TOPIC TYPES AND TAGS.

Tag	Global	Easiest	Hardest
Bad	<0.057	<0.48	<0.005
Average	0.057 ≤MAP≤0.37	0.48 ≤MAP≤0.69	0.005 ≤MAP≤0.035
Good	>0.37 (max=0.92)	>0.69 (max=0.92)	>0.035 (max=0.21)

2) Results

Figure 3 presents the results when considering the global set of topics. The categories of each qualitative variables are coded by letters. For instance, the 9 categories for the variable *Model* are coded from *"a"* to "*i*". The tree indicates that the variable *Model* is the most important (first node on top of the tree) to classify the runs. When *Model* is not "b" (BM25b0.5), most of runs get "*Average"* MAP (right child node). When *Model* is "b" (`BM25b0.5`), the next most important variable is *Field*. If *Field* is `TDN` 'label "c"), results can be "*Average*" else *Ref* becomes important to go on the classification and so on...

The results obtained in Figure 3 are consistent with the information provided by the boxplots (Figure 2). The worst results are obtained with `BM25b0.5` as the second boxplot for *Model* is clearly moved downward. The fact that the class "*Good"* does not appear in the tree is also consistent with Figure 2 as none boxplot appears above the others.

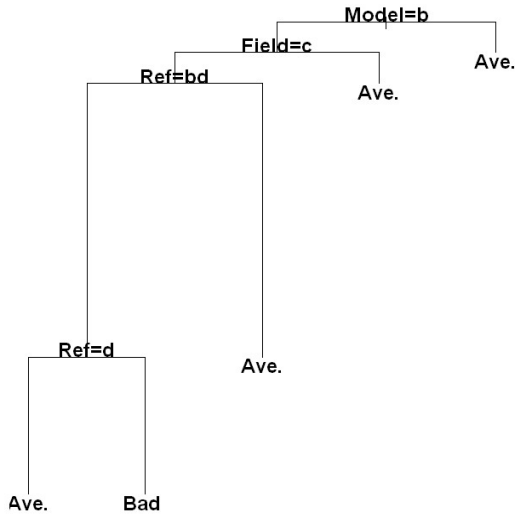

Figure 3. CART obtained with the global set of topics. The categories of each parameter are coded with letters according to the order given in Table 1.

Regarding the easiest topics (Figure 4), the parameter *Model* is still the most important in classifying the 3 categories of MAP. In addition to `BM25b0.5`, `PL2c1.0` (coded as "h") is also considered as a bad configuration. The next most important parameter is *DocNb* which provides lower values of MAP ("*Bad*" label) with `100` ("e") and `200` ("f") documents used.

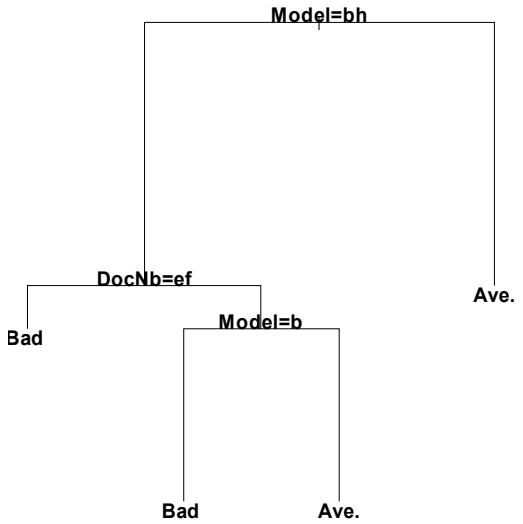

Figure 4. CART obtained with the easiest topics.

For the hardest topics (Figure 5), the structure of the tree appears more complicated although the same pruning parameters was used. Four parameters are involved: *DocNb*, *Ref*, *Model* and *Idf*. Surprisingly, *Field* does not appear in the tree. This is certainly due to potential interactions between parameters that are not taken into account with univariate ANOVAs.

VI. CONCLUSION AND FUTURE WORKS

The analysis we performed clearly confirms that some parameters produce significant changes in the performance of information retrieval systems. It also indicates that these changes are different when considering various topics characterized by unequal difficulty and provides clues to tune the parameters in order to improve the performance.

The analysis we conduced does not permit to predict performance measures but indicates the parameters that have the higher influence on the results. One important result of this study is that parameters depend on the difficulty of the topic.

This work has to be validated considering other performance measures and a more systematic procedure to characterize groups of topics.

VI. ACKNOWLEGMENTS

This work was supported in part by the ANR Agence Nationale de la Recherche (CAAS project), by the Région Midi-Pyrénées (project #10008510) and by the federative research structure FREMIT.

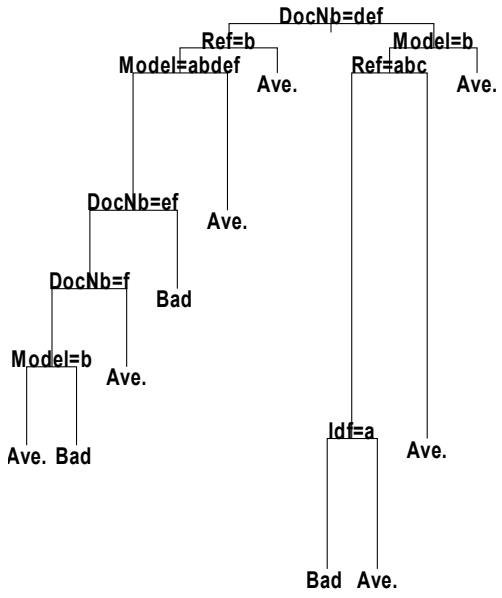

Figure 5. CART obtained with the hardest topics.

References

[1] TREC Text REtrieval Conference at http://trec.nist.gov <retrieved: 08,2011>

[2] CLEF Cross-Language Evaluation Forum at http://clef-campaign.org <retrieved: 08,2011>

[3] C. W. Cleverdon, J. Mills, and E. M. Keen, "Factors determining the performance of indexing systems. Cranfield", UK: Aslib Cranfield Research Project, College of Aeronautics, 1966, Volume 1:Design; Volume 2: Results.

[4] J. Kamps, S. Geva, C. Peters, T. Sakai, A. Trotman, and E. Voorhees, "Report on the SIGIR 2009 Workshop on the future of IR evaluation", ACM SIGIR Forum, Vol. 43 Issue 2, 2009, pp. 13-23.

[5] S. Robertson, "Richer Theories, Richer Experiments, The Future of IR Evaluation Workshop", Inter. ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, 4.

[6] D. Evans and R. Lefferts, "Design and evaluation of the clarit-trec-2 system", 2nd Text Retrieval Conference, NIST Special Publication 500-215, 1994, pp. 137-150.

[7] C. Zhai, J. Lafferty, "A study of smoothing methods for language models applied to information retrieval", ACM Transactions on Information Systems, Vol. 22 Issue 2, 2004, pp. 179 – 214.

[8] G. Salton. "The Smart Retrieval System", Prentice Hall, Englewood Cliffs, NJ, 1971.

[9] S. E. Robertson and K.S. Jones, "Relevance weighting of search terms". Journal of the American Society for Information Sciences, Vol. 27, Issue 3, 1976, pp. 129-146.

[10] S. E. Robertson and S. Walker, "Some simple approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval", Proc. ACM SIGIR conference on Research and development in information retrieval, 1994, pp. 232-241.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis". JASIST. Vol. 41, Issue 6, 1990, pp. 391-407.

[12] J. M. Ponté and B. Croft, "A langage modeling approach to information retrieval", Proc. ACM SIGIR, Conference and Research and Development in Information Retrieval, 1998, pp. 275-281.

[13] N.D. Kompaoré, J. Mothe, and L. Tanguy, "Combining indexing methods and query sizes in information retrieval in French", Proc. International Conference on Enterprise Information Systems (ICEIS), 2008, pp. 149-154.

[14] A. Lifchitz, S. Jhean-Larose, and G. Denhière, "Effect of tuned parameters on an LSA multiple choice questions answering model", Behavior Research Methods, Vol. 41 Issue 4, 2008, pp. 1201-1209.

[15] D. Harman and C. Buckley, "The NRRC reliable information access (RIA) workshop", Proc. ACM SIGIR Conference on Research and Development in Information Retrieval, 2004, pp. 528-529.

[16] D. Harman and C. Buckley, "Overview of the Reliable Information Access Workshop", Information Retrieval, Vol. 12, Issue 6, 2009, pp. 615-641.

[17] D. Banks, P. Over, and N.F. Zhang. "Blind Men and Elephants: Six Approaches to TREC data", Information Retrieval, Vol. 1, Issue 1-2, 1999, pp. 7-34.

[18] A. Bigot, C. Chrisment, T. Dkaki, G. Hubert, and J. Mothe, "Fusing Different Information Retrieval Systems According to Query Topics - A Study Based on Correlation in Information Retrieval Systems and Query Topics ", DOI: 10.1007/s10791-011-9169-5 (to appear).

[19] Terrier web site, http://terrier.org/, <retrieved: 08,2011>.

[20] Trec_eval, http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval_video/README version from the 24th july 2006, <retrieved: 08,2011>.

[21] A. Baccini, S. Déjean, L. Lafage, and J. Mothe. "How many performance measures to evaluate Information Retrieval Systems?" Knowledge and Information Systems, DOI 10.1007/s10115-011-0391-7, 2011 .

[22] L. Breiman, J.H. Friedman, R. Olshen, and C.J. Stone. Classification and Regression Trees. 1984, Chapman & Hall/CRC.

[23] W.-Y. Loh, "Classification and regression tree methods". In Ruggeri, F., Kenett, R., and Faltin, F. W., editors, Encyclopedia of Statistics in Quality and Reliability, 2008, pp. 315–323. Wiley, Chichester, UK.

[24] T.M. Therneau and B. Atkinson. R port by Brian Ripley. (2010). rpart: Recursive Partitioning. R package version 3.1-48. http://CRAN.R-project.org/package=rpart <retrieved: 08,2011>

[25] R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/<retrieved: 08,2011>.

 ISBN: 978-1-61208-162-5