A Massive Generation of IR Runs

Jonathan Louedec
Institut de Recherche en Informatique de Toulouse
UMR5505, CNRS,
118 Route de Narbonne, 31062 Toulouse, France
Jonathan.Louedec@irit.fr

Josiane Mothe
Institut de Recherche en Informatique de Toulouse
UMR5505, CNRS, IUFM Université de Toulouse
118 Route de Narbonne, 31062 Toulouse, France
Josiane.Mothe@irit.fr

Abstract -- **In this demo we present a top-level interface for the Terrier platform. Terrier is an open source search engine that implements various indexing and retrieval models from the literature. A search process can be run using various parameters for indexing and retrieval in relation to TREC and INEX collections. However, with Terrier it is difficult to generate a significant number of runs in order to evaluate the influence of the various parameters. The interface we have developed helps the user instantly generate a significant number of runs.**

Keywords –information retrieval, indexing, evaluation, runs

I. INTRODUCTION

Terrier is an open source search engine that implements various indexing and retrieval models from the literature [1]. It is very helpful for researchers in the field of information retrieval, for example, when a researcher develops a new model, Terrier is used to compare the results with the literature. It is also used in approaches needing preliminary results that are treated in more advanced ranking strategies or processes such as QA applications [2].

In this two-page summary we present an interface that we have developed in order to generate massive numbers of runs using the various models and parameters available in Terrier. Rather than having to re-write a program that requests Terrier many times in its command mode (e.g. trec_terrier.sh -r -Dtrec.model=BB2), this interface allows the researcher to decide which parameters he or she wishes to vary, in which range, etc. The interface we have developed is based on Terrier 3.5 and written in Java.

II. INTERFACE

A. *Overview*

The interface is divided into various sections (see figure 3). First the user selects the terrier directory, collection directory, qrel file and topic file. A second section allows the user to select the indexing modules he or she wishes to use. All the Terrier parameters are clickable and values can be set. A third section is dedicated to the search parameters. Finally the fourth section considers the reformulation models and makes it possible to parameter them.

B. *Range of the Parameters*

1) *Non numerical parameters*

The non numerical parameters are simply clickable items (see Figure 1). It is possible to choose several values, for example, when considering query reformulation models, multiple choices are possible. In addition the “All” choice automatically selects all the parameter values.

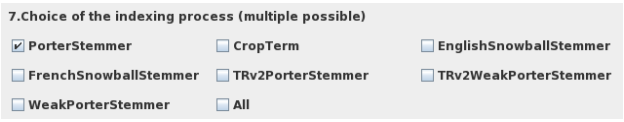

Fig. 1. Example of non numerical parameters

2) *Numerical parameters*

For numerical parameters (see Figure 2), the user provides three values: {the minimum number of documents, the maximum number and the stepped gap between the minimum and maximum numbers}. For example, if the Number of documents to be used for query expansion is set to {0;10;2}, query reformulation based on the query expansion model selected in the interface will be calculated with 0, 2, 4, 6, 8 and 10 documents. If three query expansion models are selected in the interface, then the system will compute 6 * 3 runs.

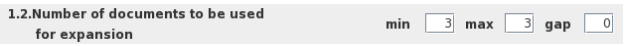

Fig. 2. Example of numerical parameter

III. GENERATING RUNS AND APPLICATIONS

A. *Exponential Number of Runs*

The number of runs that can be generated using the interface is quite high since Terrier includes many parameters (see Table 1). When considering 5 values for numerical value parameters, this gives about 6 millions

different runs (7 x 5 x 2 x 21 x 5 x 7 x 8 x 3 x 5). Other parameters, such as the stop list used and the number of indexing terms per document, have not been reported in Table I. This estimation raises a computing time issue which we have not resolved yet.

B. *Example of Applications: Analysis of Trends*

Some work has been carried out on past TREC or evaluation program results [3]. However, our interface and in particular, the resulting availability of runs, make it possible to conduct large studies of cross parameter effects, for example, in [4] we analyzed 98650 objects characterized by 8 variables. Each object corresponds to one topic processed by a chain of modules (indexing, search, query reformulation) and evaluated according to its performance. Variables consist of non redundant module parameters (topic field, indexing model, retrieving model, query reformulation model, number of documents used in query reformulation) and one performance measure (MAP) for TREC ad hoc data. We have shown that the most significant parameter is the retrieval model and that the next most significant parameter is the topic field used to generate the query.

TABLE I. Main parameters of the interface

Parameter	Number of values
Indexing models	7
Bloc size for indexing	Numerical value
Lowidf/nolowidf (indexing)	2
Weighting schema	21
Weighting parameter	Numerical value
Topic fields combinations	7
Query reformulation models	8
Query reformulation parameters	3 Numerical values

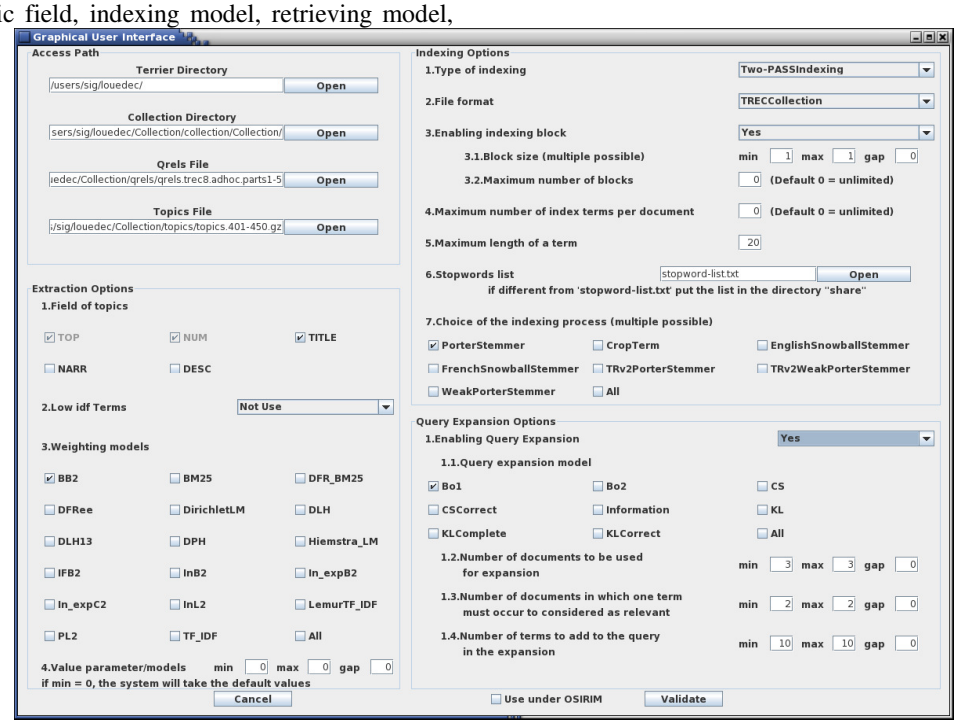

Fig. 3. Example of numerical parameter

IV. CONCLUSION

In this two-page summary we have presented a top-level interface for the Terrier platform. This interface makes it possible to generate numerous runs that are then available for conducting failure analysis and parameter influence. The next stage will be to make the program parallel so that running it on various collections will no longer be a problem.

V. Acknowledgement

We would like to thank the research federation FREMIT 3424 as well as the ANR agency (CAAS project) for their support.

REFERENCES

[1] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In Proceedings of OSIR'2006 Workshop, 2006.

[2] E. Sanjuan, V. Moriceau, X. Tannier, P. Bellot, and J. Mothe, Overview of the INEX 2011 Question Answering Track, *International Workshop of the Initiative for the Evaluation of XML Retrieval* Springer, LNCS 7424, p. 188-206, 2012.

[3] A. Bigot, C. Chrisment, T. Dkaki, G. Hubert, and J. Mothe, Fusing Different Information Retrieval Systems According to Query Topics, Information Retrieval, 14(6):617-648, 2011.

[4] J. Compaoré, A. Maïram Gueye, S. Déjean, J. Mothe, J. Randriamparany, Mining Information Retrieval Results Significant IR parameters, IMMM 2011, 156-161, 2011.