ADAPTING SEARCH ACCORDING TO QUERIES:

Experimental studies

D. Kompaoré
Institut de Recherche en Informatique
Université de Toulouse, UMR 5505 CNRS
Toulouse
France
kompaore@irit.fr

J. Mothe
Institut de Recherche en Informatique
Université de Toulouse, UMR 5505 CNRS
Toulouse
France
mothe@irit.fr

Abstract:

WWW information access is based on search engines. Most of the users consider a single search engine that they will use whatever their queries are. Additionally, current search engines follow a general framework and they treat all the queries in the same way. In this paper, we present experiments that show that considering variability both in terms of systems and in terms of queries would improve the results of a search. We based our experiments on the data from the international evaluation forum TREC (200 topics, several Gbytes of documents). This forum simulates search engine usage and gives access to topics, document collections, expected documents and system answers. We show that high precision can be improved by automatically deciding which system should be used for a given query. In our approach, system selection is based on query clustering; queries are clustered considering linguistic features that are automatically extracted from queries. We show that high precision (P@5) is improved by 8.9% for P@5 over the best system for TREC 5 for example. We also show that considering a larger set of collections, P@5 is improved of about 5%, when considering a learning/testing evaluation framework.

Keywords: contextual information retrieval, query clustering, TREC evaluation, high precision.

1. Introduction

WWW information access is based on search engines. Most of the users consider a single search engine that they will use whatever their information needs and queries are. However, recent studies showed that system's performances highly vary. Buckley and Harman (2004) consider that the comprehension of variability is complex because it is due to various parameters: the formulation of the query, the relation between the query and documents as well as the characteristics of the system. Indeed, variability can be seen as query variability or system variability. Query and system variability are the two sides of the same coin. Query variability refers to the fact that system S1 can get high performance for a query Q1 but bad performance for another query Q2 –System S2 may get opposite results. System variability refers to the fact that system S1 gets high performance for a given query Q1 and a system S2 bad performance for the same query –Considering query Q2, the systems can be ranked differently.

Despite query and system variability, most of current IR systems do not make it possible to treat differently different queries. The main exception is Question/Answering (QA) systems (Voorhees and Dang, 2006). QA systems aim at providing the users with facts as opposed to documents for adhoc information retrieval. QA systems are generally based on linguistic treatments that make it possible to distinguish various types of queries, for example considering the type of expected element (people, place, date, etc…). Knowing the type of information the user expects, QA systems adapt then the information extraction mechanism (Voorhees, 2004). Concerning the adhoc IR systems, some approaches investigate the influence of the formulation of queries or the intelligent treatments which can be carried out on this formulation. Fox and Shaw (1994) were the first to evaluate the combination of various systems by fusing retrieved document lists in different ways. Then many studies reported the impact of various form of combination (Lee, 1997), (Soboroff et al., 2001), (Lillis et al., 2006), (Kompaoré et al., 2008). Other approaches combine various query formulations and study the influence on the results (Belkin et al., 1993) (Buckley and Waltz, 2000) (Beitzel and al., 2004). They all define a general fusing function that is applied equally to any query. Finally, some studies focused on the issue of difficult queries (Allan, 2003), (He and Ounis, 2003), (Mothe and Tanguy, 2005) and how to improve results on these specific queries.

In this paper we study query and system variability. We make the hypothesis that depending on the query, different systems should be used. More specifically, we show that high precision can be improved by automatically deciding which system to use for a given query. In our approach, system selection is done on the basis of query clusters: we decide which system to use for a given query cluster; the selected system is different from one cluster to another. Queries are clustered considering linguistic features that are automatically extracted from queries.

The approach we present is evaluated on the data from the international evaluation forum TREC, using adhoc collection (200 topics, several Gbytes of documents. This forum simulates search engine usage and gives access to topics, document collections, expected documents and system answers. We show that high precision (P@5) is improved by 8.9% over the best system of TREC 5. We also show that considering a larger set of collections, P@5 is improved of about 5%, when considering a learning/testing evaluation framework

This paper is organised as follows. We first report some related works (section 2). In section 3 we illustrate query variability, first by showing how results can vary from one system to another and from one query to another. Then, by showing the linguistic characteristics of queries and the various values query characteristics can get. Given the fact that significant correlation exist, the query features have been used to cluster queries. Then, the best system for each query cluster is selected. This is presented in section 4, in addition to a comparative study that shows that high precision could be improved. More specifically, we show how high precision can be improved when selecting the best system for each cluster compared to selecting the best system. Section 5 presents other experiments to evaluate the method. It is a two phase evaluation in which a subset of queries are use to train the method and then the rest of the queries are used for testing. Comparative results are presented. Finally section 6 presents the conclusions and some future works.

2. Related works

Data fusion is related to our work in the sense they aim at combining different results. Fox and Shaw (1994) fuse different retrieved document lists. The experiments were carried out on 9 different sub-collections of TREC, and by combining 5 query formulations that retrieved 5 document lists. The authors suggested several linear combination functions. Lee (1997) complete this study and shows that the most relevant function is the so called CombMNZ for which the document score is the sum of the score it obtains in the different list multiplied by the number of list that retrieved it. Soboroff (2001) compared the use of similarity and of document rank in the fused document lists. They show that using document ranks is better than using similarity scores.

Predicting query difficulty is also related to our work in the fact that predicting query difficulty is not an end in it-self but rather a first step before answering difficult query in a different way than easy queries (Yom-Tov et al, 2004) (Mothe and Tanguy, 2005). The "Robust track" TREC task introduced in 2003 aims at trying to predict the difficulty of the queries (Allan, 2003). The results showed that it was very difficult to predict the difficulty of a query and that there is a great variability on the capacity of the systems to predict the queries which will be difficult for them (Allan, 2005). In (He and Ounis, 2003), various models of term weighting are used according to the information needs and the queries characteristics. The method showed its effectiveness on the difficult queries within the framework of the "Robust track" TREC task.

Finally query clustering is another related work. Wen et al. (2002) suggest using users' logs to cluster queries, considering the users' actions when browsing retrieved document lists. He and Ounis (2004) cluster queries according to some query features. They consider the query size (in terms of number of words), the distribution of the information carried by query terms (related to idf feature) and the clarity score (Cronen-Townsend, 2002). For each type of queries, the best system is selected. They show that the method improve the results compared to the best system.

3. Query variability

This section reports a study on query variability. Query variability reflects the fact that a system can perform well on a query and badly on another query (see examples section 3.1). The hypothesis of this study is that some inner query features can be correlated to system performance. We consider some query linguistic features. These features and how they are automatically extracted from queries are presented in section 3.2). We give some examples of exacted values in section 3.3. We then consider in section 3.4 the correlation between these features and system performance (recall and precision).

3.1 Example of query variability on TREC

Results vary from one query to another and from one system to another. A good system for a given query is not necessary good for other queries. Table 1 illustrates this variability. In Table 1, we report 5 usual precision oriented performance measures in information retrieval. Precision corresponds to the proportion of relevant documents among the retrieved documents. P@5, P@10 and P@15 or Precision at 5 documents (10, 15 documents) measures precision after 5 (10, 15) documents (whether relevant or non-relevant) have been retrieved. R-prec measures precision after R documents have been retrieved (R being the number of relevant documents) and MAP for Mean Average Precision is computed as the mean of average precisions calculated each time a relevant document is retrieved. Recall is not reported here: it corresponds to the proportion of relevant documents the system retrieves.

	Query 351	Query 378	Query 351	Query 378
	clarit98comb		ibms98a	
MAP	0.7112	0.0064	0.4553	0.1244
R-PREC	0.625	0.0204	0.5	0.1429
P@5	1	0	1	0.4
P@10	1	0	0.9	0.5
P@15	1	0.0667	0.7333	0.4667

Table 1. TREC7 System performance: different systems, different queries: different results.

Table1 shows system and query variability. For instance, the system called *clarit98comb* performs differently for query 351 and for query 378. This system gets better results for query 351 than for query 378 whatever the measure is. For example, in terms of MAP it obtains 0.7112 for query 351 and 0.0064 for query 378. The opposite result is observed when considering system *ibms98a*. If we consider system variability, for query 351, the system called *clarit98comb* performs better than system *ibms98a*. When considering query 378, systems are ranked oppositely: system *ibms98a* performs better than *clarit98comb*. This short extract of system performance shows that there is potential for performance improvement. Results would be improved if we had known that for query 351, we should have used *clarit98comb* system and *ibms98a* system for query 378. Just doing this, we could have got in average 1. as P@5 rather than 0.5 using system *clarit98comb* and 0.7 using system *ibms98a*.

The rest of the paper investigates a way to automatically decide which system to use for a given query. The decision is based on some automatically extracted linguistic features that are presented in the next section.

3.2 Extracting query features

The queries we consider are composed of three textual parts: a title that is supposed to correspond to a typical web-like user's query. It is composed of just a few words. It is written under the form of keywords and not necessary in natural language. The two other parts are written in natural language. The descriptive part explains the title whereas the narrative part describes what will be a relevant sentence and a non-relevant sentence.
The queries we consider in our experiments are richer in their information content in that they can include sentences or part of sentences in natural language and not just key-words. This is indeed richer in terms of detail than is usually available in a web search engine –and topic definitions like TREC topics can be criticised for this in some respect. However, our work is based on the premise that a user has indeed expressed his information need in such NL form.
This format of queries makes it possible to analyze the queries and to extract many features. In previous work (Mothe and Tanguy, 2005), we have described 13 linguistic features that can be extracted from a query and correspond to morphological, syntactic and semantic features. The features are as follows:

- NBWORDS: is the average length of terms in the query, measured in numbers of characters.

- MORPH: average number of morphemes per word is obtained using the CELEX morphological database, which describes, for around 40,000 lemmas, their morphological construction.
- SUFFIX: number of suffixed tokens. A bootstrapping method is used in order to extract the most frequent suffixes from the CELEX database, and then tested for each lemma in the topic if it is eligible for a suffix from this list.
- PN: number of proper nouns is obtained using the POS (Part of speach) tagger's analysis, and with a more robust method based on upper-case word forms.
- ACRO, NUM Acronyms and numerals are detected using a pattern-matching technique.
- UNKNOWN: Unknown words are those marked up as such by the POS.
- CONJ, PREP, PP: Conjunctions, prepositions and pronouns detected using POS tagging only.
- SYNTDEPTH, SYNDIST: Syntactic depth and syntactic links span are computed from the results of the syntactic analyzer. Syntactic depth is a straightforward measure of syntactic complexity in terms of hierarchy. It corresponds to the maximum number of nested syntactic constituents in the query. Regarding the Syntactic Links Span, its distance is computed in terms of number of words. We then average this value over all syntactic links.
- SYNSETS: number of meanings for a term (synsets in WordNet)

These linguistic features are extracted automatically and the query is first analyzed using some generic parsing techniques (e.g. part of speech tagging, chunking, and parsing). Based on the tagged text data, simple programs compute the corresponding information. The following tools were used:

- Tree Tagger[1] for part-of-speech tagging and lemmatization: this tool attributes a single morpho-syntactic category to each word in the input text, based on a general lexicon and a language model;
- Syntex (Fabre and Bourigault, 2001) for shallow parsing (syntactic link detection): this analyzer identifies syntactic relation between words in a sentence, based on grammatical rules;

In addition, the following resources were used:

- WordNet 1.6 semantic network to compute semantic ambiguity: this database provides, among other information, the possible meanings for a given word;
- CELEX[2] database for derivational morphology: this resource gives the morphological decomposition of a given word.

3.3 Example of query features: variability on values

Table 2 shows that query linguistic features vary. For example, query 351 is composed of 48 words if we consider TDN (Title+Descriptive+Narrative parts of the query). This query also contains 0.1250 prepositions. This number corresponds to the number of preposition the query contains, divided by the number of words of this query.

[1] *TreeTagger*, by H. Schmidt; available at www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[2] *CELEX English database* (1993). Available at www.mpi.nl/world/celex

	Query 351	Query 378
NBWORDS	48	47
NP	0.2292	0.0638
ACRO	0	0
NUM	0	0
PREP	0.1250	0.0638
PP	0	0.0638
VBCONJ	0.125	0.0851
UNKNOWN	0	0
AVGMORPH	1.1667	1.1276
SUFFIX	0.1458	0.1489
AVGSYNSETS	3.1	5.7143
SYNTDEPTH	3.25	3.6667
SYNTDISTANCE	1.64	1.3871

Table 2. Query features for 2 queries from TREC7 adhoc.

3.4 Correlation between system performance and query features

Table 3 reports the significant correlation that exists between linguistic features and system performance in terms of recall and precision. The numbers corresponds to the correlation (Pearson) and the significance (p-value). Table 3 shows that SYNTDIST and precision are negatively correlated as well as SYNSETS and recall - statistically significant (Mothe and Tanguy, 2005).

TREC Campaign	Significant variables for Recall	Significant variables for Precision
TREC 3	- PREP - SYNTDEPTH - SYNSETS (-0.302 ; 0 .033)	- SUFFIX - NBWORDS - CONJ
TREC 5		- SYNTDIST (-0.396 ; 0.000) - SYNTDEPTH
TREC 6	- SYNSETS (-0.354 ; 0.012) + PN	
TREC 7	- SYNSETS (-0.284 ; 0.045)	+ PN - LENGTH - SYNTDIST (-0.234; 0.047)

Table 3 : Significant correlations between linguistic features and recall / precision

This result clearly shows that depending on some query features, systems will perform better: the higher the SYNSET, the lower the recall will be and the higher the SYNTDIST, the lower precision will be. This preliminary result motivates the following hypothesis: it is possible to cluster queries according to their features and then to decide which system to used for each cluster. That corresponds to the study which is reported in the next sections.

4 Selecting the best system to use according to query clusters

The basic idea which is developed here is that for each query cluster, it is possible to decide which the best system to use is. Then, considering a new query, it will be possible to decide to which cluster it belongs to and thus to apply the most adapted system.

In section 4.1., we show how queries are clustered; then we show that it is possible, in most of the cases, to select the best system to use, given query clusters (section 4.2.). Moreover, we show that when considering a specific collection (TREC5 is presented), high precision is improved from 8% (P@10) to 9% (P@5) compared to official results. When considering the 4 collections, we show that high precision is improved from 2.46% (P@15) to 3.67% (P@5)

4.1 Query clusters

Query features as presented in section 3 are used to cluster queries. We consider Agglomerative Hierarchical Clustering (AHC) which provides queries clusters without any previous knowledge on the number of groups or on their structure. AHC is an iterative method in which the two closest clusters are merged at each step, starting from individuals. We use the Euclidean distance as a measure of similarity between individual pairs (query pairs) and the Ward criterion as a measure of cluster similarity. It consists in fusing the two clusters that minimize the increase in the total within-cluster sum of squares (Seber, 1984).

AHC results in a dendrogram that can be cut at different levels (see figure 1), depending on the size and inner similarity which is required. In this dendrogram, the length of the edges between nodes corresponds to the distance between them.

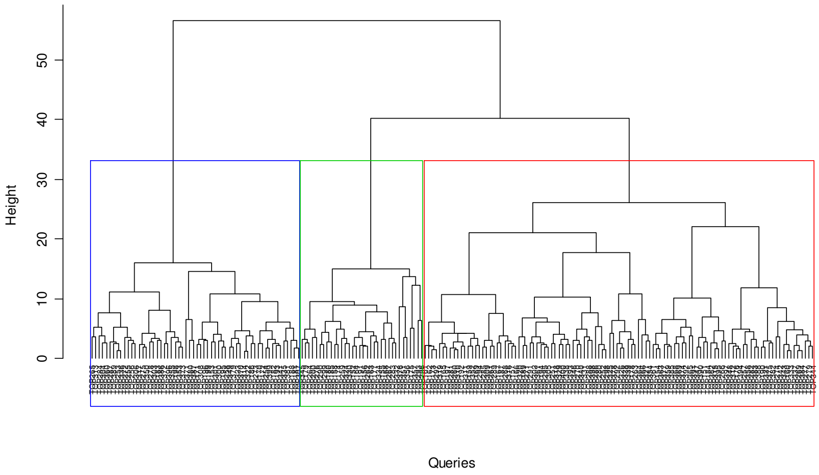

Figure 1: Dendrogram representing queries clustering.

Dendrogram from figure 1 can be cut at different levels defining a certain number of clusters. Among them 3,4,6 and 7 clusters could be obtained, mainly depending on how much cluster 3 (right side) is cut (1, 2 etc. clusters). In the reminding of the paper, we chose to cluster the queries into 3 clusters. This choice is clearly one important parameter of the study. However, this choice is motivated by the fact it optimise the distance between the obtained clusters.

4.2 Best system

In this section, we show that, given query clusters, it is relevant to select the best system. Firstly, we consider a subpart of the collection, considering TREC5 only; then we consider the all set.

Some detailed results

Table 4 reports the way TREC5 queries have been split into 3 clusters. Table 5 and 6 presents the results obtained when considering the 50 TREC5 queries and the set of systems that participate that year.

In table 5, for each measure (P@5, P@10 and P@15), we report the average performance the best system obtains considering the query cluster and compare this performance to the performance the same system obtained in average over the all set of queries. For example, considering cluster 3 and P@5, the system that obtains the best P@5 when averaged over the queries from cluster 3 is LNNaDesc2 with 0.68; this system obtains 0.50 of P@5 when considering the 50 queries. Note that overall, the best system that participates to TREC5 was Ethme1 with 0.62 (P@5), 0.55 (P@10) and 0.51 (P@15) as shown in table 6.

	Cluster 1	Cluster 2	Cluster 3
Number of queries	13	10	27

Table 4: Repartition of queries from TREC5 into 3 clusters

	Cluster 1	Cluster 2	Cluster 3
P@5	Ethme1_C 0.75 (+21%) Ethme1_all 0.62	Uwgcx0_C2 0.52 (-6%) Uwgcx0_all 0.56	LNNaDesc2_C3 0.68 (+36%) LNNaDesc2_all 0.50
P@10	Ethme1_C1 0.68 (+24%) Ethme1_all 0.55	Genrl3_C2 0.44 (-6%) Genrl3_all 0.47	LNaDesc2_C3 0.59 (+24%) LNaDesc2_all 0.48
P@15	Ethme1_C1 0.64 (+24%) Ethme1_all 0.51	Ethme1_C2 0.39 (-24%) Ethme1_all 0.51	LNaDesc2_C3 0.50 (+17%) LNaDesc2_all 0.43

Table 5 : Mean high precision of TREC5 systems in query clusters.

In table 6, for each measure, we report the value when averaging the result over the query clusters, when the best system is selected for each cluster (first column) ; the same measure is presented, but weighted considering the cluster size (second column) and the best system performance averaged over the 50 queries is reported (third column). This result clearly shows that by selecting the best system from each cluster, high precision is improved. It is improved of about 9% for P@5. In this table, as in the tables in the rest of the paper, the “best” system over the all set of queries is the one that gets the highest value of the considered measure.

	Average over the 3 query clusters (best system for each class)	Average over the 3 clusters, considering their size	Best system over the 50 query set (without query clustering)
P@5	0,6448 (+4,7 %)	0,6709 (+8,9%)	0,6160 (ethme1)
P@10	0,5700 (+4,4%)	0,5912 (+8,3%)	0,5460 (ethme1)
P@15	0,5104 (-0,8%)	0,5233 (+1,7%)	0,5147 (ethme1)

Table 6 – Comparison of mean high precision over queries in TREC5 : weather considering query clusters or not.

Results when considering the overall set

In this section, we report the same types of results, but when considering each TREC collections.

Table 7 indicates how the queries are distributed along clusters.

	Cluster 1	Cluster 2	Cluster 3
TREC3	14	15	21
TREC5	13	10	27
TREC6	16	5	29
TREC7	15	4	31
Total	58 (29%)	34 (17%)	108 (54%)

Table 7: Number of queries per query cluster for the TREC adhoc collections

Table 8 indicates the average improvement of high precision when considering the 4 collections. The first column presents the average precisions when averaged over the three clusters and the four collections, considering the best system in each cluster. The second column weights these results taking into account the cluster size. Finally the third column average the high precision obtained when considering the best system each year (without considering query clusters). Table 8 shows that P@5 is improve of about 3.67% when averaging results on the clusters (and about 4.63% when the size of clusters are considered), compared to initial results.

	Average over the 3 query clusters (best system for each class)	Average over the 3 clusters, considering their size	Best system over the set of 200 queries (without query clustering)
P@5	0.7381 (+3.67%)	0.7450 (+4.63%)	0.7120
P@10	0.6816 (+3.12%)	0.6870 (+3.93%)	0.6610
P@15	0.6407 (+2.46%)	0.6441 (+3%)	0.6253

Table 8 – Comparison of mean high precision over queries in all TREC collections, with or without query clustering.

However, in these experiments, the queries that are clustered are the queries that are used for evaluating the method. What will happen when new queries are sent? In the next section, we consider both learning and testing phases. That means that we learn what the best system is for each cluster but on a sub-set of queries. Then the rest of the queries are used to test the approach. Since the number of queries is rather small, we repeat the experiments on several learning/testing samples.

4.2 Learning and testing

The principal used in the experiments presented in this section is as follows. Some of the queries from the 200 queries are used to learn which system is the best to treat a query from a given cluster. Once the best system is known, we consider the rest of the queries (testing queries), detect the cluster they belong to and send them to the associated system. When using such a learning/testing approach, usually, the process has to be repeated a certain number of times, so that general conclusions can be drawn. We repeat the process 10 times; each time, 180 queries are used as learning queries and 20 as testing queries.

	Average over the 3 query clusters (best system for each class)	Average over the 3 clusters, considering their size	Best system over the set of 200 queries (without query clustering)
P@5	0.7479 (+5.04%)	0.7474 (+4.97%)	0.7120
P@10	0.6881 (+4.09%)	0.6877 (+4.04%)	0.6610
P@15	0.6487 (+3.74%)	0.6481 (+3.65%)	0.6253

Table 9 – Testing phase, mean high precision over queries in all TREC collections.

Table 9 reports the results averaged over the TREC collections and over the 10 stages of the experiment. The first column indicates the high precision that is obtained when averaging the measures over the clusters and the 10 iterations of the process, considering the testing queries only. In this column, for each cluster, we consider the best system for each cluster that has been learnt during the learning phase. In the second column, the results are normalized by the size of the clusters. Finally, the third column indicates the best results, without query clustering (average of the high precision over the TREC collection, considering the best system each year).

Table 9 shows that high precision is always improved. The improvement is about 5 % for P@5 when the average performance is computed on the 3 clusters and for all TREC collections used in our experiments. For P@15, the improvement is more than 3.5%.

5. Conclusions and perspectives

Query variability is analysed considering linguistic features that can be automatically extracted from the queries. We first show that some features are clearly correlated to recall and precision. We also show that it is possible to cluster queries according these features and select the best system to use for each query. Considering the best system for each cluster, we show that we can improve high precision (P@5 is improved of about 9% on TREC5 and of about 5% when considering the 4 TREC collections). When considering a learning/testing method to evaluate the results we show that high precision is still improved of about 5% for P@5. These experiments show not only that if topics are clustered, selecting a different system to treat them is efficient but also that, giving a new query it is possible to associate it to the appropriate cluster.

6. References

Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O., Goharian N. 2004. Fusion of Effective retrieval strategies in the same information retrieval system. Journal of American Society Information Science Technologies (TOIS). 55(10): 859-868.

Belkin N.J., Cool C., Croft W.B., and Callan J.P.. 1993. The effect of multiple query representations on information retrieval system performance. *16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 339–346.

Buckley C., Waltz J. 2000. SMART in TREC 8. *The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST.

Buckley, C., Harman, D. 2004. Reliable information access. Final report, *27th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Shefield: ACM Press, 528 – 529.

Cronen-Townsend S., Zhou Y., and Croft W.B.. 2002. Predicting query performance, 25th annual international ACM-SIGIR conference on research and development in information retrieval, 299–306.

Fabre, C. and Bourigault D. (2001). Linguistic clues for corpus-based acquisition of lexical dependencies, *Corpus Linguistics*, Lancaster.

Fox, E.A., Shaw, J.A. 1994. Combination of multiple searches. *2nd Text Retrieval Conference (TREC-2)*, NIST special publication, 243-252.

He B. and Ounis I.. 2003. A query-based model selection approach for the poorly-performing queries. *TREC 2003.*, 636–645.

He B. and Ounis I. 2004. A query-based pre-retrieval model selection approach to information retrieval, RIAO.

Kompaore N. D., Mothe J., Tanguy L. 2008. Combining indexing methods and query sizes in information retrieval in French. *International Conference on Enterprise Information Systems (ICEIS 2008)*, to appear.

Lee, J. 1997. Analysis of multiple evidence combination. *22th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM Press, 267-276.

Lillis D., Toolan F., Peng L., Collier R., and Dunnion J.. 2006. Probability based fusion of information retrieval result sets. *29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, 139–146.

Mothe, J., Tanguy, L. 2005. Linguistic features to predict query difficulty- A case study on previous TREC campaigns, *SIGIR workshop on Predicting Query Difficulty - Methods and Applications*.

Seber, G. 1984. Multivariate Observations. New York: Willey.

Soboroff I., Nicholas C., and Cahan P.. 2001. Ranking retrieval systems without relevance judgements. *24th Annual International ACM SIGIR Conference*, 66–73.

TREC, Available WWW: trec.nist.gov, (accessed May 2008).

Wen J., Nie J. Y., and Zhang H. 2002. Query clustering using user logs. *ACM Transactions on Information Systems* (TOIS), 59–81.

Allan J., 2003.. High Accuracy Retrieval from Documents, The Twelfth Text Retrieval Conference, p 24-36, NIST Special Publication: SP 500-255, Available WWW: http://trec.nist.gov/pubs/trec12/t12_proceedings.html (accessed May 2008).

Allan J., 2005. HARD Track Overview in TREC 2005, High Accuracy Retrieval from Documents, Available WWW: trec.nist.gov/pubs/trec14/papers/HARD.OVERVIEW.pdf

Voorhees E. M.. 2004. Overview of the TREC 2004 question answering track. Proceedings of the Twelfth Text REtrieval Conference (TREC 2004).

Voorhees E. M. and Dang H. T.. 2006. Overview of the TREC 2005 question answering track. *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005).*

Yom-Tov E., Fine S., Carmel D., Darlow A., and Amitay E. 2004. Improving document retrieval according to prediction of query difficulty, *Working Notes of Text Retrieval Conference (TREC 2004),* 393-402.

Acknowledgements

We would like to thank M. Ludovic Tanguy, Université de Toulouse, CNRS for the data he provided for this study (query linguistic features).