MaxProb and Categorization of Queries Based on Linguistic Features.

Desire Kompaore
Institut de Recherche en Informatique de Toulouse, IRIT, 118 route de Narbonne, 31062 Toulouse cedex 04, France
kompaore@irit.fr

Josiane Mothe
Institut de Recherche en Informatique de Toulouse, IRIT, 118 route de Narbonne, 31062 Toulouse cedex 04, France
mothe@irit.fr

ABSTRACT

In this paper, we study the use of a probabilistic fusion approach inspired from the probFuse algorithm [14]. Our fusing technique is based on queries that are classified using some automatically extracted linguistic features [11]. In this approach, performances which are estimated during a training phase are used as an indicator of the systems to fuse. The rank list of relevant documents provided by the systems are divided into segments and used in a training/testing process to detect the systems to fuse. Our fusion technique shows better performances than other fusion techniques like CombMNZ [10], ProbFuse [14] and Best_sys [11].

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering, Search process, Retrieval models.

General Terms

Algorithms, Measurement, Performance, Reliability, Experimentation.

Keywords

Linguistic features, classification, probability, fusion, Information retrieval, queries, retrieval systems.

1. INTRODUCTION

User's satisfaction is a central element in the Information Retrieval (IR) process. IR systems try to find the most relevant documents after the user has expressed his need. System performances are of high variability; for instance, a system A can get better performance than a system B for a given query, but lower performance for another query. Understanding these variabilities is complex due to various parameters like the query formulation, the relation between the query and the documents, and finally the characteristics of the IR systems [3].

In our approach, we consider that analyzing the query content could help in "understanding" the query and thus in satisfying the user's need. Rather than using only the so called "bag of words" representation, Natural Language Processing (NLP) features could extend systems capabilities in the retrieval process. Generally speaking, NLP attempts to reproduce the human interpretation of language by assuming that the patterns in grammar and the conceptual relationships between words in language can be articulated scientifically. A simple example of linguistic techniques applied to IR is to expand queries using term synonyms and meronyms [6]. Many other linguistic elements can be considered in a query. For example, in [17] the authors have investigated the effect of linguistic feature on predicting query difficulty. The conclusion was that some linguistic features are correlated to the poor performance of some systems, and that a number of correlations exist between these features and the traditional recall and precision measures.

In this paper, we consider both query variability in terms of query formulation and system variability. Our hypothesis is that if system A is more efficient on a certain type of queries than system B and reversely, then system B can be better than system A on another type of queries. Type of queries is viewed in our approach as clusters of queries.

In this paper, we investigate the use of a probabilistic-based fusion approach, applied on a set of queries that are clustered by automatically extracted features. Systems are evaluated on each cluster of queries and during a training phase, each system is associated with a probability score. This score is an indicator of the system performances for subsequent queries. Using these probabilities, we compute data fusion for a number of testing queries on each query cluster. We compare the results we obtained with CombMNZ method [10].

This paper is organized as follows: we present in section 2 the related works. In section 3 we present the data sets we have used in our experiments and the linguistic features extracted from the queries. Section 4 presents the method which is based on query clustering and system selection with a probabilistic approach. Section 5 presents preliminary results based on an extract of the data. Section 6 evaluates the method on the four years of TREC in training and testing mode. Section 6 concludes the paper.

2. RELATED WORK

A lot of variability types exist in a retrieval process from query expression to retrieval techniques. The Reliable Information Access (RIA) workshop [3] investigated the reasons why system

Conference'04, Month 1–2, 2004, City, State, Country.

performance varies across queries. They analyzed the failures on TREC topics and found 10 failure categories [2]. One of the conclusions of that workshop was that "comparing a full topic ranking against ranking based on only one aspect of the topic will give a measure of the importance of that aspect to the retrieved set". Indeed, some of the failures are due to the fact that systems emphasize only partial aspects of the query. Variability of queries has been studied by [5] in their experiments in TREC-8 query track, giving 4 versions of each of the 50 TREC-1 topics. [5] has studied the variation on the performances when short or long queries are used; the results show that short queries perform better than longer ones. Query variability has also been studied in [1], combining multiple query representations in a single IR engine, and thus comparing this combination with other combinations. The authors show that the query length normalization method they've suggested outperforms traditional data fusion approaches. [7] focus on automatic query expansion methods that construct query concepts from the documents space. The authors extract features from the document space for clustering. The results of the clustering then correspond to primitive concepts that are considered as basic elements of the query concepts.

Query difficulty is another track: the hypothesis is that, even if system variability exists, some queries are difficult for almost all the systems. The concept of clarity score has been proposed as an attempt to quantify query difficulty [8]. The clarity score has been shown to be correlated positively with the query average precision and thus can be considered as a predictor of query performance. [8] use documents in order to predict query difficulty using a clarity score that depends on both the query and the target collection. [17] show that two linguistic features are correlated to recall and precision (syntactic links span for precision and polysemy value for recall; both negatively). System variability has also been studied. In this case, the variability is relative to the retrieval techniques used by the systems. A number of works in the IR literature study how to use system variability to improve retrieval efficiency. [10] investigate the effect of data fusion on retrieval effectiveness. The authors show that merging results from many retrieval systems improve performance compared to the one obtained by a single retrieval system. In [10], it is shown that the best combination method (CombSUM) consists in making a sum of all the similarity values for all the documents. Lee [13] uses the CombMNZ method (that in addition to CombSUM considers the rank each document gets in the fused systems) and studies the level of overlap between relevant and non relevant documents. This study shows that it is better to fuse systems with high overlap of relevant documents.

In this paper, we consider that combining systems can help in improving efficiency. However, in our approach, rather than fusing different results, we select among a set of systems the most relevant system to use. The selection is based on clusters of queries that take into account variability and homogeneity in query features.

3. INPUT DATA

3.1 Collections and measures

We use TREC adhoc data in our experiments. TREC adhoc collections consist in a set of documents (more than 3 Gb in TREC-3 for example), a set of 50 queries, and the runs (retrieved documents for each of the queries) of all the participating systems to the TREC campaign. Such a collection exists for different years. *Trec_eval* is used to evaluate systems performances according to different measures (recall, precision, MAP, R-precision and P@). We use TREC-5 data for our experiments. Different systems have been in competition in TREC- campaigns (80 systems for TREC-5).

In addition to being an international benchmark, TREC collections are ideal for evaluating our hypothesis because systems are evaluated on the same basis, and queries are long and can be analyzed to extract linguistic features. We present in the next section the linguistic features used in this experiment.

3.2 Linguistic features

We present in this section the linguistic features that were automatically extracted and used in our experiments.

Queries in TREC contain a title, a description and a narrative part. The format of TREC queries allows a linguistic analysis on these queries. In the work of [17], 13 linguistic features have been extracted for each query and we use these features as a basis for query clustering. The features can be described 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 was used in order to extract the most frequent suffixes from the CELEX database, and then tested for each lemma in the topic if it was 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 polysemy value (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 Tagger1 for part-of-speech tagging and lemmatization: this tool attributes a single morpho-syntactic category to each word in

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

the input text, based on a general lexicon and a language model;
- Syntex [9] 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;
- CELEX2 database for derivational morphology: this resource gives the morphological decomposition of a given word.

The linguistic features can be put in 3 main groups: morphological features, syntactic and semantic features. In the next section, we indicate how we use these features to classify queries.

4. USING LINGUISTIC FEATURES TO CLUSTER QUERIES

Clustering is a well known technique used in IR. Most of research tries to group documents with same characteristics into the same cluster.
In this study, clustering is done at the beginning of the retrieval process: the query. Linguistic features are automatically extracted from queries and are used to classify the queries. The objective is to group queries that have very close linguistic features. The resulting clusters are used to analyze systems' performances and detect the best system or combination of systems to use for each cluster. We use the agglomerative hierarchical clustering method to classify queries.

4.1 Query representation and clustering method

Query representation

Linguistic features are presented in a 200x13 matrix in which queries stands for individuals and linguistic features for variables. Each query can be viewed as a vector initially located in a 13-dimensional space, each dimension corresponding to one linguistic feature. Similarly, each linguistic feature can be viewed as a vector in a 200-dimensional space, each dimension corresponding to a query. Data were previously centred and reduced according to columns in order to homogenise variables. More information about these methods used to pre-treat data can be found in [12].

Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering (AHC) provides queries clusters without any previous knowledge on the number of groups or on their structure. It can be viewed as an iterative algorithm that starts with the configuration "Each individual is a cluster" and finish with "All individuals are joined into a single cluster". During each iteration, the two closest clusters are merged. The AHC requires defining the following elements:
- A measure to compute distance between each pair of individuals.
- An agglomeration criterion that can be viewed as a distance between two clusters.
The most popular choice for the former is the Euclidean distance. Moreover, this choice is natural when reduced data are analyzed. Euclidean distance between 2 queries X and Y is defined as:

$$d(X,Y)=\sqrt{\sum_{i=1}^{13}(X_i-Y_i)^2}$$

For the latter, the Ward criterion is generally advocated by statisticians. It consists in fusing the two clusters that minimize the increase in the total within-cluster sum of squares [18]. Results of AHC can be represented as a tree or dendrogram which nodes correspond to the union of 2 individuals or 2 clusters (see figure 1).

4.2 Clustering results

As said previously, Euclidean distance and Ward criterion were used in the analysis.

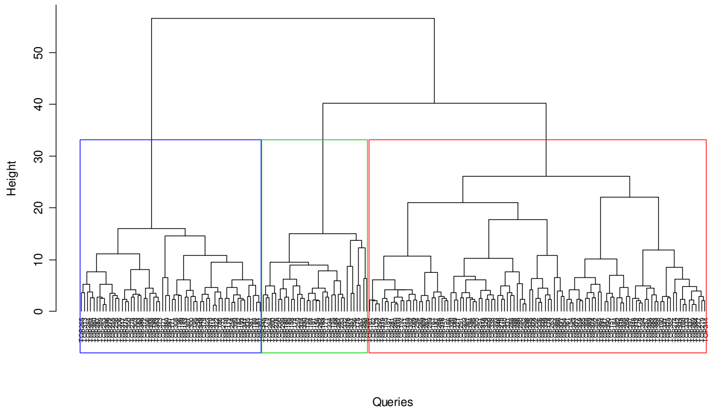

Figure 1: Dendrogram representing queries clustering.

Three groups of clusters are used for the remaining of the experiments [11].
After clustering query with AHC, we applied k-means methods (initialized with the centroid of each cluster determined by the AHC) in order to stabilize query clustering. We will not present in this work the k-means method; it is detailed in [12].
Query clustering was computed on the entire set of 200 queries from various TREC sessions. However, runs from different sessions (TREC year) are independent; that is to say that a given system provides retrieved documents only for one session. As a result, before evaluating systems, we filter the queries according to the TREC session they belong to. The following table gives the repartition of queries for each TREC session and for each group of queries. |Q| represents the number of queries per class.

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

Table 1: TREC5 collections characteristics

	|Q|	TREC5
Cluster1	58 (29%)	20,69%
Cluster2	34 (17%)	29,41%
Cluster3	108 (54%)	25%

Table 1 shows that, for example, cluster 3 contain 54% of the total number of queries and that 25% of these queries come from TREC5 data. The numbers in this table correspond to the average of all the 10 training phase.

Given these 3 clusters of queries, we evaluate system performance for each query cluster. We use the *trec_eval* program used to calculate the performance of all the systems. Trec_eval evaluate the systems with a certain number of measures. We are interested in this work in high precision measures (P5, P10 and P15) which correspond to the precision after 5, 10 or 15 documents are retrieved.

4.3 Probabilistic fusion

In this section, we describe the probabilistic approach used in our experiments.

This experiment was inspired by the probFuse approach presented by [14]. ProbFuse is a probabilistic fusion approach that uses estimated probabilities to calculate the score of a document. During a training phase, systems accuracy to retrieve relevant document is estimated by a probability score. During a testing phase, the score of the documents retrieved by the retrieval models is calculated by taking into account the probability issued by the training phase. The testing and training phase are done on a set of queries.

A retrieval system (retrieval model) retrieves a set of documents as a result of the user's query. Some works in the literature rank documents in the results set and use this ranking to evaluate systems' performance. According to the probFuse approach, rather than using only this ranking, the resulting ranked list of documents provided by each system are divided into x segments.

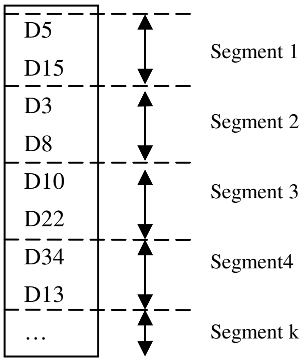

Figure 2: ranked list of relevant documents retrieved by system s for query q.

In figure 2 each of the 4 segments contains 2 documents. Documents in segment 1 are more relevant than documents in segment 2 or later. ProbFuse calculates the probability that a document belonging to a segment k is relevant for a given query. This probability is calculated for each segment of the ranked list of documents for each query and each system. The probability score is then averaged over the total queries that are available.

In a training set of Q queries, P(d_k/m) is the probability that a document *d* retrieved in segment *k* by the retrieval system *m* is relevant. This probability is calculated as follow:

$$P(d_k / m) = \frac{\sum_{q=1}^{Q} \frac{|R_{k,q}|}{k}}{Q}$$

where $|R_{k,q}|$ is the number of documents in the segment *k* that are judged as relevant for the query *q*, and |*k*| is the total number of documents in segment *k*. In figure 2, |*k*| = 2 .In the remaining of the experiment, we have fixed the number of documents per segment to 25 (|*k*|=*25*) according to previous experiments on probFuse. The authors find that 25 is an optimal value for the number of documents in a segment. This parameter will be tuned in other experiments to analyze its impact on the results.

In a second phase (testing phase), a ranking score is calculated for all the retrieved documents. This score is calculated according to the following formula:

$$S_d = \sum_{m=1}^{M} \frac{P(d_k|m)}{k}$$

where M is the total number of retrieval models being used, $P(d_k|m)$ is the probability of relevance for a document *d* retrieved in segment *k* (k=1 for the first segment, k=2 for the second segment, and so on) by the retrieval model *m*. According to this score calculation, any retrieval model that does not return document d in its result set at all is associated a probability score of zero ($P(d_k|m)$ =0) in order to avoid any boosting of the document ranking score from systems which do not return document d.

Our experiments are done on TREC-5 collections. The main process of our experiments can be depicted as follows:

Training phase

```
- Extract linguistic features of queries
- Cluster queries with these features
- Choose the segment size (|k|)
- For each cluster
     - For each query
          - For each retrieval system (m)
               - Calculate for each segment P(d_k/m)
     - Calculate max P(d_k/m) for each segment
```

Testing phase

- Associate each testing query to the best cluster

```
- For each cluster
    - For each query
        - For each system
            - For each segment
                - Use the system with max $P(d_k/m)$ for retrieval in segment k
        - Create a fused ranked list relevant documents for all the segments
        - Evaluate this list with trec_eval
    - Compare the results with other techniques
```

We choose to test in this study the first step of ProbFuse. For all the queries in the training phase, the maximum value of the probability is calculated for each segment and the corresponding system(s) is used to retrieve the documents in the segment where its gets the maximum probability value. Suppose for instance that we define a size of 3 for all the segments. Let s1 be the system that gets the maximum probability score during the training phase for segment 1 and segment 2, and s1 the best system for segment 3. In our experiments, the resulting fused list is composed of the top 6 documents (ranked from position 1 to 6) of s1 and documents from position 7 to 9 of s2. We evaluate afterward the efficiency of our results with trec_eval. We call our method MaxProb. Next section gives the preliminary results obtained during our experiments.

5. EXPERIMENTS AND EVALUATION

Experiments presented in this section are done on TREC-5 data. In figure 3, *MaxProb* algorithm is compared to 3 other fusion techniques. The baseline is the best system of TREC-5 (system with the higher P5, P10, and P15 measures).

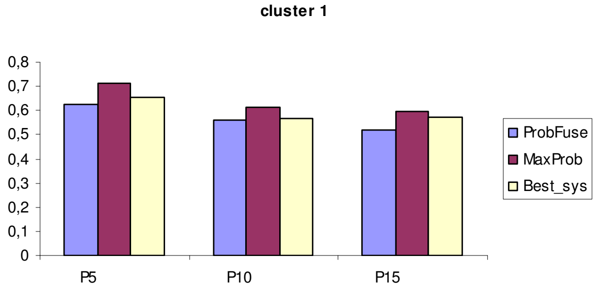

Figure 3: Local comparison of different fusion techniques on cluster 1.

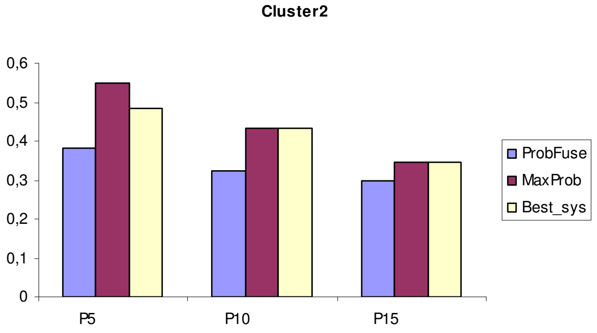

Figure 4: Local comparison of different fusion techniques on cluster 2.

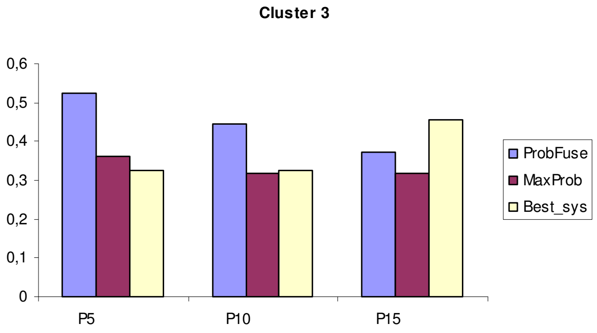

Figure 5: Local comparison of different fusion techniques on cluster 3.

In figure 3 and 4, *MaxProb* achieves better than *ProbFuse* or our previous *Best_sys* method for all the high precision measures. This local comparison evaluate if *MaxProb* gets the same global efficiency on each of the clusters. Contrary to figure 3 and 4, figure 5 shows that *ProbFuse* is better than *MaxProb* and *Best_sys* for all the measures. The reason of this under performance is that for cluster3, in average, the systems chosen by *Best_sys* and *MaxProb* fail to retrieve relevant documents for the test queries. *ProbFuse* take advantage of the fusion process to improve its performances.

We present in the following a global comparison of *MaxProb* and other fusion techniques.

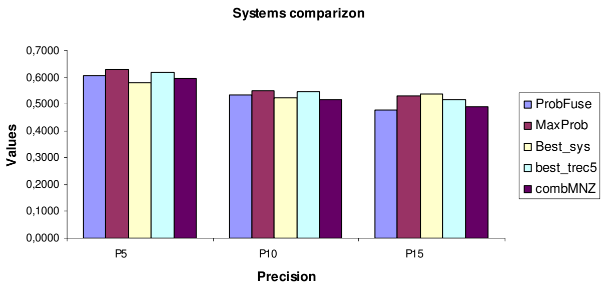

Figure 6: Global comparison of different fusion techniques.

The general results on high precision show an improvement of performance when *MaxProb* is used compared to the other methods. For the P5 measure, *MaxProb* achieves better than all the other techniques. We can see in the figure that for P5, *ProbFuse* and best system of TREC-5 get the same performance for the P5 measure. The 2 other techniques *Best_sys* and the traditional *CombMNZ* technique also get the same performance.

6. CONCLUSION

In this experiment, the size of each segment is 25, and performance is evaluated for high precision measures (P5, P10, and P15). The conclusion of this study is that our method is better than some the original *ProbFuse* algorithm, traditional *combMNZ* method and finally achieves better than the best system of TREC-5. The good results obtained by *MaxProb* show that partitioning the result list into segments has a positive impact on the results. Therefore, *ProbFuse* algorithm also uses segmentation of results before calculation of scores. The main difference between these two methods seems to be the number of system used in the fusion. *ProbFuse* doesn't fuse all the score of the systems. In TREC, differences between top systems are very small, but can be very important compared to bad systems. Therefore, *MaxProb* selects only the best system for each segment for the fusion process. It may be profitable to select the number of systems to fuse and this will probably improve the results. On going works propose a variant of the ProbFuse algorithm combining both ProbFuse and MaxProb methods. The same conclusion can be drawn for combMNZ algorithm. We are also interested on determining the impact of the size of the segments on the results and some of our work is going in this direction. Collection may also have an impact on the result. Future work will also try to apply *MaxProb* on other collections and with other measures.

7. REFERENCES

[1] 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. J. Am Soc. Inf. Sci. Technol., 55(10): 859-868

[2] Buckley, C. (2004): Why current IR engines fail In Proceedings of the 27th annual International ACM SIGIR conference. ACM Press, 584-585

[3] Buckley, C., Harman, D. (2004): Reliable information access. Final report, 27th International ACM SIGIR Conference. Shefield: ACM Press, 528 - 529

[4] Buckley C., Waltz J. (2000): SMART in TREC 8. In The Eighth Text REtrieval Conference

[5] (TREC-8). Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST.

[6] Buscaldi, D., Rosso, P., Sanchis Arnal, E. (2005): A WordNet-based Query Expansion method for. Geographical Information Retrieval, CLEF, http://www.clef-campaign.org/2005/working_notes/workingnotes2005/buscaldi05.pdf.

[7] Chang, Y., Kim, M., Ounis, I. (2004): Construction of query concepts in a document space based on data mining techniques. Proceedings of the 6th International Conference On Flexible Query Answering Systems (FQAS,2004). Lecture notes in Artificial Intelligence, Lyon, France, June 24-26, 137-149

[8] Cronen-Towsend, S., Zhou, Y., Croft, W.B. (2002): Predicting query performance. Proceedings of the 25th annual international ACM-SIGIR conference on research and development in information retrieval, Tampere, 299-306

[9] Fabre, C., Bourigeault, D. (2001): Linguistic clues for corpus-based acquisition of lexical dependencies, in Proceeding of Corpus Linguistics, Lancaster

[10] Fox, E.A., Shaw, J.A. (1994): Combination of multiple searches. Proceedings of the 2nd Text Retrieval Conference (TREC-2), NIST special publication, 243-252

[11] Kompaore .D, Mothe.J, Baccini.A., Dejean.S. Prediction du SRI à utiliser en function des critères linguistiques de la requête. In Coria 07, Saint Etienne, pp. 239-254

[12] Lebart, L., Morineau, A., Piron, M. (2006): Statistique exploratoire multidimensionnelle : Visualisations et inférences en fouille de données. 4ème édition, Dunod, Paris, 6 juillet 2006

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

[14] Lillis. D. Toolan.F. Peng. L. Collier. R. and Dunnion. J. Probability-based fusion of information Retrieval Result sets. in Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, Seattle 2006.

[15] Mandl, Womser-Hacker, T. (2003): Linguistic and statistical analysis for the CLEF topics. Peters C, Braschler M, Gonzalo J and Kluck M, Eds. Advances in Cross-Language Information Retrieval: Results of the CLEF 2002 Evaluation Campaign, LNCS 2785. Spinger Verlag, 505-511

[16] Mardia, K.V., Kent, J.T., BIbby, J.M. (1989): Multivariate Analysis. Academic Press.7th Printing.

[17] 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.

[18] Seber, G.: Multivariate Observations. New York: Willey (1984)