Inf Retrieval (2011) 14:617–648
DOI 10.1007/s10791-011-9169-5

Fusing different information retrieval systems according to query-topics: a study based on correlation in information retrieval systems and TREC topics

Anthony Bigot · Claude Chrisment · Taoufiq Dkaki · Gilles Hubert · Josiane Mothe

Received: 20 July 2010 / Accepted: 12 May 2011 / Published online: 21 June 2011

Abstract To evaluate Information Retrieval Systems on their effectiveness, evaluation programs such as TREC offer a rigorous methodology as well as benchmark collections. Whatever the evaluation collection used, effectiveness is generally considered globally, averaging the results over a set of information needs. As a result, the variability of system performance is hidden as the similarities and differences from one system to another are averaged. Moreover, the topics on which a given system succeeds or fails are left unknown. In this paper we propose an approach based on data analysis methods (correspondence analysis and clustering) to discover correlations between systems and to find trends in topic/system correlations. We show that it is possible to cluster topics and systems according to system performance on these topics, some system clusters being better on some topics. Finally, we propose a new method to consider complementary systems as based on their performances which can be applied for example in the case of repeated queries. We consider the system profile based on the similarity of the set of TREC topics on which systems achieve similar levels of performance. We show that this method is effective when using the TREC ad hoc collection.

Keywords Information retrieval · Local analysis of results · Dimensionality reduction techniques · Correspondence analysis · Clustering · Hierarchical clustering · System/query correlation · Query clustering · System clustering · Meta search

A. Bigot · C. Chrisment · T. Dkaki · G. Hubert · J. Mothe (✉)
Institut de Recherche en Informatique de Toulouse, UMR 5505, CNRS, Université de Toulouse, 118 Route de Narbonne, 31062 Toulouse Cedex 04, France
e-mail: mothe@irit.fr

A. Bigot
e-mail: bigot@irit.fr

C. Chrisment
e-mail: chrisment@irit.fr

T. Dkaki
e-mail: dkaki@irit.fr

G. Hubert
e-mail: hubert@irit.fr

1 Introduction

To evaluate information retrieval (IR) systems on their effectiveness, international evaluation programs such as TREC,[1] CLEF[2] or INEX[3] offer a rigorous methodology as well as benchmark collections. These evaluation programs have contributed enormously to the IR field (Kamps et al. 2009; Robertson 2009). Consider TREC ad hoc for example (Harman 2000): in TREC 2 and 3, new term weighting functions were introduced with Okapi, BM25 (Robertson and Walker 1994). The same year blind relevance feedback was introduced (Evans and Lefferts 1994) and is now used widely for automatic query reformulation; in TREC 3, the merging of system results was introduced (Fox and Shaw 1994); and in TREC 7 multiple uses of passages were considered (Allan et al. 1998). These results can be clearly linked to the evaluation program and are now widely used in information retrieval. TREC has gathered the responses to a survey on the uses of TREC resources and estimates of the value of TREC resources to the participants' organization (Rowe et al. 2010). Even if evaluation programs do not cover all the aspects of IR evaluation, the importance of these international programs is unquestioned by the IR community. In addition to having made important contributions in the IR field, these programs and the associated benchmark collections have other advantages for research groups. The first advantage of these collections is that the evaluation process is easier, as there is no need to build a new document collection, associated queries, and relevance judgments from real users. Building an evaluation collection is time and resource demanding for academics who have no access to search engine query logs. The second advantage is that results can be directly compared to other published work without requiring re-implementation of different approaches, because results obtained from different systems also become known from various publications.

A benchmark collection from international evaluation programs is usually composed of a set of documents, a set of topics (information needs from which a query will be built to be submitted to the search engine) and a set of document relevance judgments. In addition to the collection, a framework is provided, including a task to achieve, and measures to evaluate the systems that participate in the task. One of the characteristics of evaluation programs is that system performance is computed in a global way, by averaging measures (e.g. recall, which measures the percentage of relevant documents the system retrieves, precision which measures the percentage of retrieved documents that are relevant and derived measures) on a set of topics. Although this principle allows system ranking or comparison, detailed analysis is not presented in the program reports and any differences in system results that might exist are unknown.

For example, Table 1 reports MAP (mean average precision) over the 50 topics for a TREC 7 ad hoc search task (see Sect. 3 for details) for some of the best systems. In addition, the last line indicates the best system when considering the measure. These results can be obtained using trec_eval on official TREC runs.

From Table 1, it can be concluded that CLARIT98COMB and t7miti1 have similar results according to MAP and are indeed the best systems according to this measure. This measure is computed considering several points in the retrieved document list; precision is computed each time a relevant document is retrieved, then the average of the precisions (AP) is calculated; MAP is the mean of APs over topics. Table 1 also reports for some topics detailed results for the same runs where only AP (average precision) is considered.

[1] TREC Text REtrieval Conference—http://trec.nist.gov.

[2] CLEF Cross-Language Evaluation Forum—http://www.iei.pi.cnr.it/DELOS/CLEF.

[3] INEX Initiative—http://inex.is.informatik.uni-duisburg.de/.

Table 1 MAP and AP for some official TREC runs and selected topics

Run	MAP	AP			
	All topics	T372	T380	T391	T359
CLARIT98COMB	0.3702	0.5555	0.3594	0.1870	0.0299
t7miti1	0.3675	0.8061	0.1264	0.0760	0.0125
CLARIT98CLUS	0.3525	0.5202	0.2455	0.1962	0.0252
Ok7ax	0.3033	0.1399	0.3832	0.5261	0.0254
Brkly26	0.2905	0.2891	0.0700	0.2780	0.0586
Best system	0.3702	0.8061	0.4845	0.5261	0.1567
Average AP		0.1991	0.1893	0.1888	0.0157

CLARIT98COMB and CLARIT98CLUS are two runs submitted by CLARITECH Corporation. Ok7ax is one of the runs submitted by Okapi Group (City U./U. of Sheffield/Microsoft) using Okapi system and T7miti1 is one of the runs submitted by Management Information Technologies, Inc.

The line 'Best AP' reports the best AP obtained for each selected topic (these can come from different systems for the different topics).

Detailed results clearly show that on average AP is very close for topics T380 and T391 (last row). However, CLARIT98COMB, which is the best system on average, fails on T391 and performs relatively well on T380 (even if it is relatively far from the best system for this topic). t7miti1 outperforms the other systems on topic T372. Regarding T359, the best systems all fail, even if, when looking at the entire set of results, one can find several systems that manage to get AP of about 0.15. This is not by chance as those systems get MAP between 0.2 and 0.3. CLARIT98COMB and t7miti1 which were very close when considering MAP (Table 1) are now quite far apart regarding the results they obtain for the considered topics, and the former is closer to CLARI98CLUS in terms of profile. By system profile we mean numerical representation. A system representation consists of the list of performance values it obtained for each topic (a line vector of the system/topic table—Table 1).

From this short example we can see that global results may hide interesting differences on runs that could be used to better understand why systems fail or succeed for some topics. Alternatively, this knowledge could be used to decide which system would be better to use for each topic. Because evaluation is carried out globally, by averaging the results over about fifty topics (if we consider most of the tracks of the different international evaluation programs), the added value of the different techniques used, such as natural language processing (NLP), stemming, relevance feedback… to name a few, is not easy to demonstrate. In the context of global evaluation, it is not possible to discover correlations between topic and system successes or failures. This is the core point of this paper in which we apply data analysis techniques to discover these types of correlation and comment on them. This paper however can also be viewed as a contribution towards longer term objectives such as:

1. Defining a typology of queries and discovering the processes that should be applied to optimize the results on each type of query,
2. Building a system as an intelligent agent that would automatically decide the process required according to the type of query the user submits, including for example, the most appropriate query reformulation method or the best ranking on a per topic basis.

These are long term objectives we do not claim to achieve in this paper. The work we present is rather to be seen as a first contribution toward the construction of such an intelligent system.

Immediate objective is to analyze the data that results from evaluation programs and to define a new method of combining systems that take into account the findings resulting from the analysis. In November 2000, the DENIS-P J104814.7-395606 star was identified while mining the old DENIS database (1975, 1986); we think that, like in astronomy, new knowledge can be extracted from past results and used in IR. More precisely, in this paper, we consider detailed analysis of the results (we chose the TREC ad hoc track) and correlate them with topics. The data analyzed corresponds to traditional evaluation measures that result from TREC evaluation and which are available for each topic individually, even though not widely used (as we said, effectiveness is usually considered globally, not on a per topic or query basis). The objective of the analysis is to have more ideas on how information is correlated and structured and to use this knowledge to define new IR techniques.

The specific objectives of the paper are to:

1. Cluster topics that behave in the same way and comment on these clusters,
2. Cluster systems that behave in the same way and comment on these clusters,
3. Analyze topics and systems simultaneously and discuss their correlation,
4. Define a new method that considers various systems and select them on a per query-basis.

The paper is organized as follows: In Sect. 2 we report related work which includes data fusion, variability and past result analysis. Section 3 presents the framework of the study: the IR tasks and test collections we chose, some mathematical background. It also describes the method we propose to cluster topics and correlate topic clusters with system performance. Section 4 reports the analysis of system results when applying the method presented on TREC 7 ad hoc data. Section 5 presents a system combination technique based on the findings and discusses the improvements observed. After a conclusion (Sect. 6), an "Appendix" provides some additional details of the results.

2 Related work

In depth analysis of IR systems' retrieved lists and performance is useful for discovering information that could help to understand the influence of different IR components on retrieval performance or to improve information retrieval mechanisms. For example, results could be improved if it was possible to decide which parameter setting, system, or system combination would work in a given context. Data fusion, variability analysis and past result analysis are work that intends to tackle this challenge. For this reason they are described in this section.

2.1 Data fusion

Data fusion relies on the fact that different strategies lead to different results and thus merging these results into a single result list may improve retrieval performance. Fox and Shaw (1994) were the first to show that combining the results from multiple retrieval runs improves retrieval performance over any of their individual retrieval methods. Experiments have been carried out using 9 different TREC sub-collections, combining 5 individual runs

in different ways. The same system, but with different topic representations (queries) was used. The CombSUM (each document receives a score which is the summation of the set of similarity values obtained by this document for the individual runs) was found to be the most effective combination. Other methods have been proposed such as the Borda count (Aslam and Montague 2001), user oriented approaches (Elovici et al. 2006; Hubert et al. 2007), high precision oriented approach (Hubert and Mothe 2007), methods that combine positive and negative feedback (Dkaki and Mothe 2004), or data fusion considering lists retrieved using various document parts (Hubert and Mothe 2009).

Various studies tried to explain data fusion results (Lee 1997; Beitzel et al. 2003). Ng and Kantor (2000) associated the effectiveness of fusing two systems to two predictive variables: similarity of retrieval performances and dissimilarity between the retrieval schemes approximated as a dissimilarity between their ranked list outputs.

Wu and McClean (2006) have analyzed deeper correlations between the systems to be fused. As opposed to previous approaches, the analysis is topic-based and leads to different weights to define the contribution of each fused system (according to topics). Correlation in this work is based on the (ranked) list of retrieved documents and corresponds to generalization of CombSUM. The results report a slight improvement compared to Comb-like methods (CombSUM, CombMNZ and other variants). In the context of multimedia information retrieval, Wilkins et al. (2006) also investigated the use of a weighting fusion that is topic-dependent.

More recently, learning to rank has been used to automatically learn the best ranking function from training data (Trotman 2005; Cao et al. 2007). In learning to rank, examples to learn in the training phase consist of ranked lists of the relevant documents associated with the corresponding topics. The testing phase uses the unique learned ranking function but on new topics.

2.2 Variability, topic difficulty and topic clustering

Variability in results is one major issue to understand data fusion and system combination. Understanding this variability can lead to better fusion strategies. Harman and Buckley (2004, 2009) claims that understanding variability in results (system 1 working well on topic 1 but poorly on topic 2 with system 2 doing the reverse) is difficult because it is due to three types of factor: topic statement, relationship of the topic to the documents and system features. The "reliable information access" workshop (Harman and Buckley 2004) focused on the query expansion issue and analyzed both system and topic variability factors on TREC collections. 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.

One subfield of variability is topic variability. The case of difficult queries and topics has been specifically studied. Mandl and Womser-Hacker (2003) analyzed CLEF topics and the correlation between topic features and system performance. They found a correlation of 0.4 between the number of proper nouns and average precision. Cronen-Townsend et al. (2002) introduced the clarity score which is a measure to predict query difficulty. This score is based on the relative entropy between the query language model and the corresponding collection language model. Mothe and Tanguy (2005) analyzed TREC topics according to 13 linguistic features and showed that the average polysemy value of topic terms is correlated with recall. Carmel et al. (2006) showed that topic

difficulty depends on the distances between three topic components: topic description, the set of relevant documents, and the entire document collection.

Another issue is to consider topic and query clusters in order to adapt the system to the query type. In (He and Ounis 2003) topic characteristics were used to cluster the topics. The best term-weighting schema in terms of precision/recall measures is then associated with each topic cluster. After this training, when a new topic is submitted, it is clustered into the existing clusters and the pre-trained system is used to process it. Different term weighting schemes are applied depending on topic characteristics. When applied to TREC Robust track, this method improves results on poorly performing topics. Kurland (2009) consider query-specific clusters for re-ranking the retrieved documents. Clustering queries has also been studied in the web context (Wen et al. 2002), but based on query logs, which is a different issue. Mothe and Tanguy (2008) have shown that topics can be clustered according to their linguistic features and that these features are correlated with system performances.

2.3 Past results analysis

Fine-grain analysis of detailed published results (Systems/Topics/Measures) has also been carried out. The study in (Banks et al. 1999) reports different analyses of TREC data. In the analysis which is most related to ours, they consider a matrix in which rows and columns represent systems and topics; cells correspond to average precision. This matrix is used in order to analyze the clusters that could be extracted. To do so, they used hierarchical clustering based on single-linkage. This method combines clusters that minimize the distance between closest elements in each. The figures included in the paper hide the distances between clusters; in addition the paper neither discusses the obtained tree nor the clusters that could have been extracted by cutting the tree at different levels. A more detailed analysis, including going back to the raw data might have helped to extract more knowledge from the obtained graphs. Our contribution goes a step further.

A different type of analysis is reported in (Mizzaro and Robertson 2007). One aim of this work was to identify a small number of topics that would be useful to distinguish effective and ineffective systems. The paper uses, as we do, effectiveness measures on system-topic pairs. The paper concludes that their experiments confirm the hypothesis that "some systems are better than others at distinguishing easy and difficult topics". The paper also concludes that "a really bad system does badly on everything, while even a good system may have difficulty with some topics" and "that easiest topics are better at distinguishing more or less effective systems". However, they do not appear to use these findings in further studies. In Chrisment et al. 2005, TREC runs are analysed in order to visualize correlations between systems and topics.

In this paper we describe an approach based on data analysis methods (correspondence analysis and clustering) to discover correlation between systems, between topics and to find trends in topic/system correlations. We also use the results of this analysis to promote a new system combination technique that makes use of both topic clusters and system clusters and show that MAP is significantly improved.

3 A method to analyze past runs

The motivation of our work is that the same unique system cannot handle properly any given query. The search for a universal IR method is probably vain; our hypothesis is that there are more benefits to expect from specialized IR approaches. In addition, analyzing

system variability could help to know which queries or which query types a given system treats best.

We aim first to analyze in depth the results of various systems, and second, to use the findings from the analysis to develop a new system combination technique that would leverage the best of each system.

3.1 TREC as an experimental test collection

For the analysis, it is compulsory to consider various systems based on the same collections (information needs, documents on which the search is carried out and relevance judgments). International experimental environments, such as TREC, accumulate such retrieval results with a large variety in terms of systems, tasks, and test collections. Systems themselves are not available, but the results they produced are. Choosing a suitable test collection is of utmost importance to the proving of soundness and usefulness of an information retrieval system.

An evaluation collection consists of the following:

- A number of pre-defined documents (e.g. newspaper articles),
- A set of topics (each topic is transformed into a query that is submitted to a given system), and
- The list of relevant information items corresponding to each topic (referred to later as qrels).

Both topics and relevant sets are manually defined. Relevance judgments are used to measure system performance by comparing relevance judgments and 'runs' (lists of documents a particular system retrieves for each topic of the test).

As several systems use the same test collection, a set of runs and their corresponding performance are available as a result of experimental environments. Our choice goes naturally to this type of resource.

Because it is impossible to analyze all the results collected from international evaluation programs, we rely on the conclusions of the SIGIR workshop "the Future of IR Evaluation" to pick-up TREC as a test collection. Indeed, one conclusion of this workshop is that the experimental paradigm Cranfield set and TREC perpetrates had lead to huge advances in the field and are still useful (Robertson 2009). We made the decision to focus on data from the TREC ad hoc retrieval task. The ad hoc task was introduced at the beginning of TREC in 1992. "The ad hoc task investigates the performance of systems that search a static set of documents using new questions (called topics in TREC)" (trec.nist.gov). 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. Each TREC participant submits runs to NIST that are evaluated against human judgments. Details on the set of documents and topics can be found at trec.nist.gov.

We chose the TREC 7 collection since it had enough participants (103 runs submitted for the 50 topics—topics 351–400) for the analysis to be meaningful.

The detailed runs a participant submits from previous TREC campaigns are available on the TREC server in a login/password protected area (http://trec.nist.gov/results.html). Notice that each run may consider a different topic representation for the same topic; the query used by each system is unknown and in this study we consider the systems as black boxes. The performance measures obtained for each topic by each run can be computed by the official TREC evaluation software known as trec_eval (which includes 135 individual measures). When evaluating a run, a measure is first computed for each topic and then

averaged over all topics. Table 1 (see Sect. 1) provides some examples of run results (average results over the set of topics).

In this paper, we considered mainly Average precision (AP) and Mean Average Precision (MAP) to evaluate system performance. For AP, the precision is calculated after each relevant document is retrieved. These precision values are then averaged together to compute AP. MAP averages AP over topics and was first introduced as a measure in TREC 2. It is an interesting measure because it aggregates recall/precision curves in a single measure (it combines different measure points). MAP is less dependent on the number of relevant documents than are high precision measures, for example. This measure is used for global comparisons of different systems (Voorhees 2007).

Runs and evaluation of these runs are the inputs of the analysis we report in Sect. 4.

3.2 Mathematical background

We propose to analyze system results using both clustering methods and correspondence analysis. These methods have been used in various domains including social sciences (Greenacre and Blasius 1994), textual data mining (Lebart et al. 2006), and genomics (Tekaia et al. 2002).

Both types of techniques can be applied on a matrix of data, for example on a matrix (denoted X) composed of n observations or elements that constitute the population to be analyzed and that is described along p variables using numerical values. Such a matrix can also be viewed as a set of n vectors in a p dimensional space. In our specific case, the matrix we analyze is composed of n TREC topics described along p systems using AP (see Table 2; Sect. 5.1).

3.2.1 Clustering methods

Clustering methods are used for representing proximities among the elements through subsets or clusters. Two major types of clustering have been defined in the literature (Lebart et al. 2006): *hierarchical clustering* which aims at defining a hierarchy of clusters partially nested in one another and *partitioning methods* that lead to partitions of the elements.

Table 2 Extract of the AP matrix from TREC 7 ad hoc

	APL985L	APL985LC	APL985SC	AntHoc01	Brkly24	Brkly25
T351	0.2257	0.2261	0.1655	0.2933	0.2987	0.3137
T352	0.0229	0.0321	0.0594	0.0277	0.0379	0.0097
T353	0.3271	0.3052	0.2852	0.2091	0.374	0.264
T354	0.1119	0.1496	0.0908	0.0139	0.0192	0.1084
T355	0.0973	0.0688	0.0327	0.1365	0.0987	0.183
T356	0.052	0.0593	0.0462	0.0091	0.0128	0.0452
T357	0.1358	0.1803	0.1391	0.0984	0.3284	0.3277
T358	0.0994	0.0988	0.0489	0.1514	0.2078	0.3887
T359	0.0378	0.0337	0.0146	0.0223	0.0319	0.0357
T360	0.39	0.3825	0.4096	0.0404	0.3275	0.036

A hierarchical clustering (HC) produces a set of partitions, $P1$, …, $Pn - 1$, Pn, of the initial elements, using a bottom-up approach. The hierarchy can be represented as a tree structure or dendrogram. At one extreme, Pn (the leaves of the tree) consists of n single elements. At the other extreme, $P1$ (the root of the tree) corresponds to a single group that consists of all n elements. In such a clustering, at each particular stage, the two clusters which are closest together are joined to form a new cluster. At the first stage, each cluster is composed of a single element. The clustering ends when a single cluster is obtained. Hierarchical clustering implies defining the distance between two elements and between two clusters. With regard to the former distance, Euclidean distance is usually used. With regard to the latter, statisticians often suggest using a method known as the Ward criteria (Ward 1963), which consists of minimizing the within-class variance of the partition.

Considering the n elements (observations) to cluster as points in a Euclidean space according to the p variables (dimensions), HC based on the Ward criteria in a Euclidean space is defined by the two following formulas:

– once two clusters C and C' of centers γ_C and $\gamma_{C'}$ are clustered together as C'', the coordinates of the resulting new center $\gamma_{C''}$ is defined as:

$$\gamma_{C''} = \frac{(m_c\gamma_c + m_{c'}\gamma_{c'})}{(m_c + m_{c'})} \tag{1}$$

– the dissimilarity between C and C' reflects the consented effort to achieve their agglomeration and is defined as:

$$\mathrm{Dis}(C, C') = \frac{m_c m_{c'}}{m_c + m_{c'}} \|\gamma_c - \gamma_{c'}\|^2 \tag{2}$$

where m_c is the mass of cluster C defined as the cluster cardinality.

We used this method since it has the advantage of being compatible with correspondence analysis that we also use for data mining. Notice that a clustering method can be performed indifferently on the rows or on the columns of the initial matrix; we can thus use it to cluster systems or topics or both (Madeira and Oliveira 2004).

HC computes a hierarchical representation of the dataset. However, most of the time, it is desirable to define clusters to analyze the results. To achieve this, the resulting hierarchical representation can be cut at any level, considering one partition of the elements. The lower the level, the fewer the number of objects per cluster and the larger the number of clusters is. An example of a dendrogram resulting from HC and a cut at one level is shown in Fig. 1. Finding good cuts is not a trivial action. Ferraretti et al. (2009) propose an automatic technique for cluster creation in HC based on the fact that resulting clusters should be compact, separated and balanced. However, the usual method used to decide the cutting level remains manual, by analyzing the distance between the nodes in the dendrogram (see Sect. 5.1 also). Seber (1984) suggests stabilizing the resulting clusters by applying a partitioning method on the clusters obtained using hierarchical clustering. K-means (MacQueen 1967) is a partitioning method that aims at partitioning n observations into k clusters, while minimizing the within-cluster sum of squares. The general algorithm uses an iterative method: an initial set of k means is first specified randomly, each observation is then assigned to the cluster with the closest mean, and finally the new means of the observations of the clusters are calculated (centroids). The process is repeated using these new k means. The algorithm stops when there is no more modification in

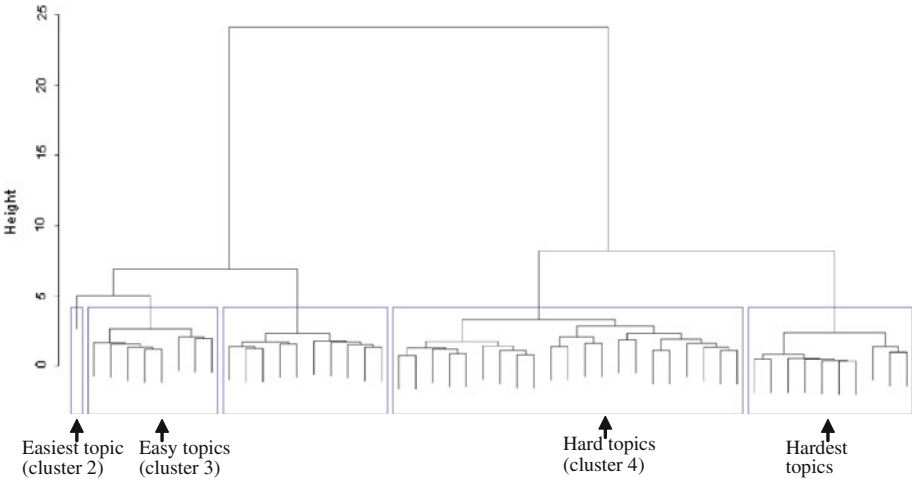

Fig. 1 Dendrogram resulting from topic clustering using AP on TREC 7 ad hoc. The detailed composition of the clusters is provided in the "Appendix"

element assignments to the clusters. When combined with HC, the number of clusters of the partitioning (k) is set by the cut chosen when using hierarchical clustering. The k means are then computed considering the elements that are clustered in this initial partition. Then, the partitioning method assigns each element to the nearest cluster (using its distance to the centroid). The algorithm stops when no element changes between two successive stages.

We use this combination of HC and K-means in the analyses presented in Sect. 5.

3.2.2 Correspondence analysis

Dimensionality reduction techniques correspond to a set of methods that aim at representing a set of n observations initially represented according to p observed variables in terms of q unobserved variables (factors).

Principal Components Analysis (PCA) and Correspondence Analysis (CA) (Benzécri 1973) are similar methods; the latter being generally less known than the former. The general goal of these data analysis methods is to represent observations, initially in a space of p dimensions (variables), in a space of lower dimensionality (Jolliffe 2002). Both methods are mathematically related to singular value decomposition. They reduce the number of data dimensions, retaining the most important, as determined by the greatest eigenvalues of a square symmetric matrix resulting from the initial matrix (Murtagh 2005). In PCA, this matrix is often a correlation matrix or the variance/co-variance matrix. In CA, the square matrix is the matrix of profiles. The eigenvectors corresponding to the highest eigenvalues are then known to be the most useful for visualizing the maximum amount of information. Moreover, the most specific information will be displayed first.

To go a step further, in the initial matrix, some variables may be different in value. When considering PCA, to enforce homogeneity, the observations are centered. The matrix to be singular value decomposed is $\left[x'_{ij}\right] = \left[x_{ij} - \bar{x}_j\right]$ where $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ and x_{ij} is the jth value of the ith observation; this transformation modifies the origin of the data. The matrix X^tX to be diagonalized is the covariance matrix (Lebart et al. 2006). Observation

homogeneity is also reinforced considering standard deviation (data is reduced) in order to enforce similar intervals on all variables (Murtagh 2005). The matrix to be submitted to singular value decomposition is then the correlation matrix Y^tY of the p variables where

$$Y = [y_{ij}] = \left[\frac{x_{ij} - \bar{x}_j}{\sigma_j}\right] \quad \text{and} \quad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}$$

(Lebart et al. 1998).

Unlike PCA, CA treats rows and columns in an identical manner whereas PCA would give priority to either the rows or the columns. In CA, the matrix to be submitted to singular value decomposition is:

$$x'_{ij} = \frac{f_{ij} - f_{i.}f_{.j}}{\sqrt{f_{i.}f_{.j}}} \quad \text{where} \quad f_{ij} = \frac{x_{ij}}{x},\ x = \sum_{ij} x_{ij},\ f_{i.} = \sum_{j} f_{ij},\ \text{and} \quad f_{.j} = \sum_{i} f_{ij}$$

In CA, if two observations (rows) have identical profiles (e.g. identical or proportional), they can be combined without affecting the variables (columns). The same is true for two identical profiles of variables.

In this method, after diagonalization, a set of eigenvectors is produced; it is then possible to project the n row-points onto the principal factors. The factors correspond to the eigenvalues of the variance/covariance matrix. The eigenvalue corresponds to the variance of one factor, that is, the variance of the coordinates of the observations according to the corresponding factor. A row-point or a column-point *contributes* to the total inertia explained by each factor. Inertia is the generalization of the variance to a multidimensional space. Graphical representation of the data is generally based on the factors that have the highest variance, the ones that explain the data best. A main advantage of CA compared to PCA is the fact that simultaneous representation of row-points and column-points is made possible (Lebart et al. 2006). Notice that CA can also be viewed as a 2-mode or biclustering approach, based on visualization (Murtagh et al. 2000). Even if the proximity between a given row-point and a column-point cannot be interpreted, the relative position of one row-point can be interpreted. Generally, a 2-D representation based on the two factors that correspond to the two highest eigenvalues is considered; however it can be useful to visualize the next principal factors since it can happen that the first factor corresponds to obvious information. Column-points and row-points can be arranged in this 2D-space. Reading the graphical representation of a CA according to some of the principal factors is not necessarily obvious and some reading keys to interpret will be provided along with the figures in Sect. 5. One important point is that column-points that appear on the periphery of the virtual hyper-sphere indicate interesting variables. They depict a vector that can be explained by the observations that contribute the most to this vector (i.e. whose projection is maximal relative to this vector). This is illustrated in more detail in Sect. 5.3, when showing the results of our analysis.

There is another reason that leads us to choose CA to analyze the AP matrix. PCA is usually used for a matrix that describes observations according to variables using quantities. PCA will show the main dispersion. It is not the most appropriate in the case of the data we want to analyze since we aim at discovering topic and system profiles. CA is generally applied in the context of contingency tables. A 2-D contingency table typically describes a population of individuals according to two variables. The rows of the matrix correspond to the modalities of one variable and the columns the modalities of the second variable. Modalities are assumed to be independent. The matrix describes the frequencies of the modalities in the population, so that the sum of the values of one row (resp. column)

is equal to the total population. CA is applicable when the sums of the rows and the sums of the columns are meaningful. In the case of the matrix we want to analyze, the sum of the values in a row (resp. column) makes sense: it is the sum of AP: the higher this value, the easier the topic (resp. the better the system). Row profiles and column profiles make sense as well. The matrix we analyze is not a contingency table but contains scores between 0 and 1. Avila and Myers show that CA can be used to "for the analysis of data matrices where the variables are measures on a ratio-scale" (Avila and Myers 1991). Those reasons lead us to consider CA when analyzing our data.

3.3 Analyzing TREC results

The goal of the analysis we propose is to discover the structure of the results of search systems when applied to a set of topics for ad hoc searching: are there some systems that behave the same way? Are there topics for which systems behave the same way? Can we learn more from the system effectiveness results? To answer these questions, we developed a scenario based on the effectiveness matrix (n topics-observations as rows, p systems-variables as columns and AP as coordinates). This scenario is composed of two steps: clustering and CA:

- Topics are first clustered according to the results that systems obtained in the TREC runs. More precisely, when studying AP, the topic/system AP matrix (Table 2) is analyzed applying HC; systems are analyzed the same way.
- Secondly, the same matrix is analyzed using CA. Two graphical representations are then made: the first one represents the variables (systems) in a reduced space. The second one represents the observations (topics) in the same reduced space. In addition a color is associated with each topic cluster extracted from the previous step. Systems and topics are then analyzed simultaneously.

Lebart et al. (2006) recommend combining CA and clustering methods mainly to enrich the representation from a multidimensional point of view:

- Using clustering corrects the distortions that can occur when using CA because of the projections in a reduced space,
- Even if this is a reduced space, CA considers a continuous space; a continuous space is more difficult to describe than a discontinuous space that produces clustering.

To be combined, clustering methods and CA should be using the same metric. Ward's hierarchical clustering and CA are both based on inertia and thus will provide coherent results in our analysis. It makes sense to use them together since they provide different visualizations and can lead to complementary interpretations as we will see in Sect. 5. It is also important to remember that HC can be applied to cluster either rows or columns and that CA treats rows and columns in an identical manner.

4 Analyzing topic and system performances

Clustering is an optimisation task that aims to assign of a set of objects into clusters so that objects in the same cluster are similar and objects from different clusters are dissimilar.

Finding what makes some objects belonging to a giving cluster and what kind of properties they share—beyond the fact that they are similar according to a giving similarity measure—goes beyond clustering goals. In the scope of clustering it is enough to find

clusters among a set of objects but succeeding in effectively interpreting the clusters is undeniably a plus. Unfortunately, interpreting clusters can present some difficulties. This is generally the case of unsupervised learning algorithms that tend to produce complicated results that may not be interpreted. When interpretation is needed, it is necessary to consider supervised learning methods such as rule induction (Liu 2006). This is not the purpose of our method.

It is also difficult to find sound interpretation for factorial axes (factors) in AC. Indeed, a combination of variables does not usually find a straight forward interpretation.

In the following, each time it is possible we will interpret the clusters and factorial axes we extracted or display. Otherwise, we will simply stick to the fact that the systems that belong to the same cluster treat each query in the same ways, and that the queries belonging to a same cluster raise the same degree of difficulty, either absolute or relative, to each system.

4.1 Topic clustering: TREC 7 ad hoc: AP

A first analysis consists in clustering the topics using HC. In such an analysis, topics play the role of observations and systems the role of variables. The data analyzed is the AP obtained by each system for each topic (see Table 2). We use the cluster package (http://cran.r-project.org/web/packages/cluster/index.html).

We applied HC to the data and obtained the dendrogram presented in Fig. 1 in which the length of the branches indicates the distance between clusters. Considering the matrix we analyzed, two topics are considered as close to each other and thus clustered together if, when considering any system, they get a similar AP. From this clustering, after applying K-means, a few changes take place. The detail of the cluster content is not to be read in the dendrogram itself, the final groups of topics are obtained after K means and are presented in the "Appendix 1".

Cutting the dendrogram at a given level corresponds to defining a partition, each cluster being defined by the list of elements it contains. The cutting level is decided considering the distance between nodes. To be relevant, a cut should occur where there is a large gap between the distances of two consecutive dendrogram nodes. When considering the tree presented in Fig. 1, we can choose a 2-cluster partition. The next relevant pruning is at 5 clusters (which is marked in Fig. 1).

The clustering itself does not label the clusters. In our specific case, labeling can be made going back to the initial matrix and having a look at the values. When considering the resulting 5-cluster partition, and after reordering for readability purpose, the second cluster on the left side (Fig. 1) corresponds to the easy topics for the systems: the topics for which the mean AP over the systems is the highest (the first cluster on the very left side consists of one topic; this is the easiest topic). This second left-side cluster (see "Appendix" for cluster content after K-means) consists of topics which have the highest AP (after the easiest topic T365 that constitutes the first cluster) when averaged over systems (from 0.3370 to 0.4628; 0.3908 on average). The cluster on the very right side of Fig. 1 corresponds to the hardest topics. The 11 topics from this cluster obtain the poorest AP (averaged over systems), varying from 0.0242 to 0.0981, 0.0493 on average for the topics from this cluster. Topic clustering results in grouping together topics according to their level of difficulty on average.

These topic clusters will be used later on and combined with system analysis.

4.2 System similarity based on AP for TREC 7 ad hoc

In the same way as for topics, it is possible to cluster systems. Considering the same data as in Sect. 5.1, we clustered the systems using HC. Figure 2 displays the resulting dendrogram. Considering this analysis, systems are considered as similar if they obtain similar results for each topic. In that way, two systems are close to each other because they both fail or succeed on the same topics. K-means has also been applied and the detail of the cluster content is listed in the "Appendix 2".

Figure 2 and cluster contents shows that versions of the same system tend to be very close. For example, the three versions of the CLARIT98 system are grouped together in the same cluster (first cluster on the left of Fig. 2). Notice that CLARIT98 runs are among the best runs and for these runs MAP is between 0.3351 and 0.3702. The same thing occurs with Okapi versions that are grouped together in cluster 2. The fact that different versions of the same system tend to yield similar performance is not that surprising since generally, different versions of systems correspond to parameter tuning. This shows that one cannot expect high enhancement of IR system performance by just making minor changes. This, in a way, supports the conclusion from (Croft 2000) considering data fusion: fused systems should be independent of each other. This is also one of the results that provide motivation for the combination technique we develop in Sect. 6. A closer analysis of the results presented in Fig. 2 reveals some other interesting insights:

- Reading the publication associated with the runs CLARITECH has submitted (trec.nist.gov), it happens that CLARIT98COMB is a combination of several other runs CLARITECH submitted. The results show that it was not a very effective combination (there is no important variation of the results). One conclusion could be that for the systems to be fused effectively they should be independent from each other.
- IRIT labs sent three runs using the Mercure system; two belong to the same cluster (third cluster from the left side), but one belongs to the last cluster. This suggests that the versions of the system lead to significant modifications, which was the case indeed. Statistical tests applied to the results show this as well.

Given the way the clustering is done, mathematically, the clusters are not necessarily reflecting the average performance over topics. Rather, the clusters reflect homogeneity on

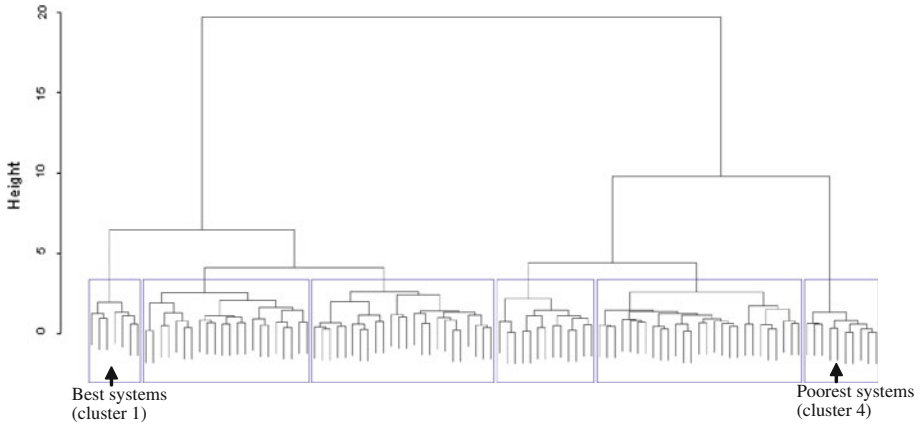

Fig. 2 Dendrogram resulting from system clustering using AP matrix from TREC 7 Ad hoc. Details of the clusters are provided in the "Appendix 2"

performance they get for individual topics. However, a closer look at the initial matrix and data shows that:

- The first cluster on the left (Fig. 2) contains the best system with regard to MAP (CLARITY98COMB, 0.4548). On average, the systems from this cluster obtain 0.3451 for MAP, compared to 0.1992 when considering all the systems. This cluster consists of 7 systems. The data also shows that this cluster contains the best ranked systems (ranks 1–6 and 8 when considering MAP). The system ranked the 7th is ok7ax which belongs to another cluster.
- The systems that have the lowest performance are grouped together in the 4th cluster from the left in Fig. 2. These 11 systems obtain the lowest MAP which is on average 0.0451. They correspond to the systems ranked from 93 to 103 when considering MAP.

These two elements show that the most effective systems tend to behave the same way, as do the systems that achieve poor results. Note that *behave* means perform relatively better/worse/equal on the same topics. Considering the other clusters, MAP is not uniformly distributed and thus cannot be used to label the clusters. In the "Appendix 3", we show the systems ordered by increasing MAP and the clusters they belong to.

These observations go in favor of the development of techniques that would be topic dependent and that would consider types of systems. Indeed, we found that some systems (that we grouped together) tend to behave the same for the same topics: either succeeding or failing; but that other groups of systems do not behave the same on the same topics. These observations are the starting point of the method developed in Sect. 5.

4.3 Further analysis of topic-system correlation: TREC 7 ad hoc: AP

Another way to analyze how systems (and topics) behave, compared to each other, is to consider CA. The visualizations associated with CA display the distances among topics and among systems (see Sect. 3.2). We used the R ade4 package (http://pbil.univ-lyon1.fr/ade4/).

Figure 3 displays the two first principal factors that correspond to 31.3% of the total inertia; only systems are represented. Factor 1 corresponds to 20.5% of the total inertia: this means that the systems are first distinguished considering this factor. The two first factors explain about 1/3 of the information. There are 25 factorial axes (factors) and each vertical bar shows the amount of information carried by the corresponding factor (see *Proportion%* on the left side). The dashed curve expresses the cumulative proportion of inertia accounted for by the factorial axes and shows that—for example—exploring the five first factors will give an overview of about half the total information (see *Cumulative proportion* on the right side).

In Fig. 3, the arrangement of column-points (systems) according to the two factors can be read as follows: the higher their values, the higher their contribution to the factor (either positively or negatively). When analyzing a factor, one should consider the coordinates (the contribution) of the point to the factor. In the case of Fig. 3, displayed points correspond to systems. Systems on the right of the graph contribute positively to factor 1.

For example CLARIT98XX, itt98ma1, t7miti1, and umwt7aXX are the systems that contribute most positively to factor 1. These systems are among the best when considering MAP. On the contrary, systems that are on the left of factor 1 are systems that contribute negatively to factor 1 (e.g. jalbse013 and KD7XX systems). They are *opposite* to the previous systems considering this factor. Indeed, going back to the initial matrix, these systems are among the ones that achieve the poorest MAP (ranks 99, 100, and 102). Factor 1 can be labeled as *Poorest systems* on the left side and *Best systems* on the right side.

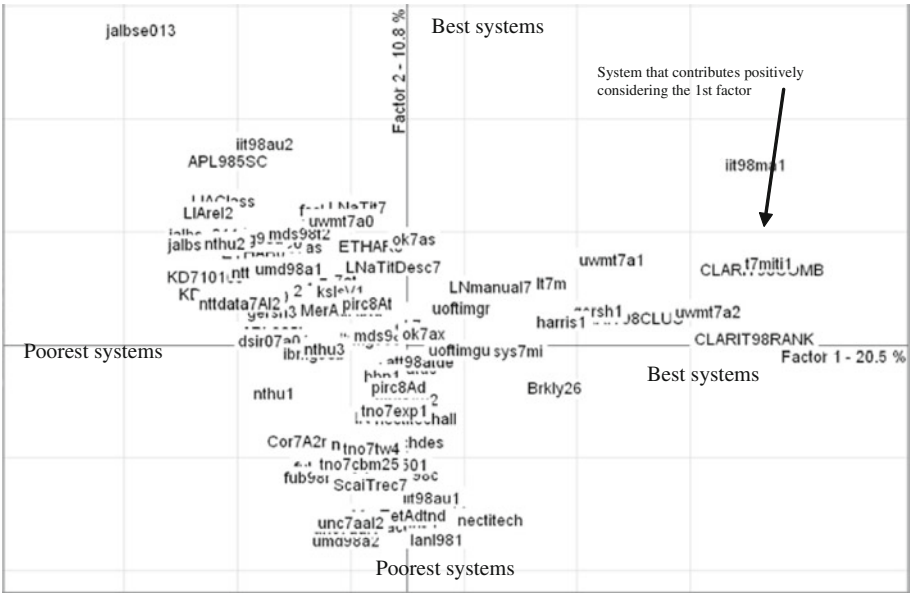

Fig. 3 The 2 first factors of the CA using the AP matrix on TREC 7 ad hoc. Systems are displayed on the 2D space induced by the two first factors which correspond to 20.5 and 10.8% of the total inertia. Some elements are hidden by others due to the fact they are too close each other

It is important to note that not surprisingly Fig. 3 is overlapping in content with Fig. 2. For example, the three versions of the CLARIT98XX systems are close to each other (right side of factor 1). They are also close to t7miti1 and uwmt7a2. CLARIT98XX, itt98ma1, t7miti1, and umwt7aXX belong to the same cluster (see Fig. 2 or "Appendix 2"). Additionally, when applying PCA on the same data (data are first centered on observations), results remains coherent. For example, CLARIT98XX and umwt7aXX close to each other and KD7XX systems in a different region of the space. However, PCA does not allow one to explain, in a simple way, relationships between variables and observations. For this reason, we display CA results only.

It is interesting to note the specific arrangement of ok7ax (near the centre of Fig. 3). This system obtains rank 7 (over 103) considering MAP; that means it is among the best performing systems. HC did not cluster it with the other best systems (e.g. CLARIT98XX systems). The results we obtain with CA are consistent since ok7ax and CLARIT98XX are distant when considering factor 1. Since both are good systems (considering their performance in terms of MAP), one can conclude that they are not good for the same reasons (not the same topics).

These two observations (ok7ax and—let us choose—CLARIT98COMB are among the best systems with regard to MAP and the two systems do not behave the same way since they are not clustered together) would suggest that these two systems could be combined in some way, for example on a per topic basis. This is another motivation of the method we propose Sect. 5.

Considering factor 2, the systems that contribute the most in a positive way are: jalbse013, iit98au2, APL9858C, LIAXX, LNaTit7 … On the other hand, ic98san4, Ianl981, umd98a2, iit98au1 are among the systems that contribute the most oppositely. Ic98san4 (out of screen shot in Fig. 3) is on the negative part of factor 2; it is placed among the poorest systems according to MAP (86th rank). Factor 2 can be labeled *Poor systems* on

the bottom part and *Good systems* on the top. Both HC and CA show that they do not fail because of the same topics. They do not belong to the same clusters in the HC and they do not contribute the same way to the main principal factors. For example, KD7XX contribute negatively to factor 1 whereas Ianl981 contribute negatively to factor 2.

When analyzing the other factors, some other systems appear as having a specific behavior. For example, Factor 3 (see "Appendix 4") clearly shows the cluster that contains ok7XX systems (cluster number 2 in the "Appendix" and second cluster from the left in Fig. 2).

An interesting feature of CA is that we can simultaneously visualize the column and the row points (systems and topics in our case). As row and column projections are related by a barycentric relationship (Murtagh 2005), it is possible to define distances in Euclidean space and display both observations and variables in the same graphical representation. Since there are many points, we prefer to show two figures. First, Fig. 4 displays topics only. It is important to understand that Figs. 3 and 4 use the same two first principal factors. Then, Fig. 5 displays the centroid of each system cluster (the 6 clusters that are displayed in Fig. 2), rather than displaying the 103 systems, in addition to the 50 topics.

Figure 4 presents the results of the CA based on the AP matrix, displaying row-points/topics only. It is displayed according to the previous factors 1 and 2; remember they correspond to more than 30% of the total inertia and have been labeled according to the MAP system obtained. Topics that contribute the most to the principal factor 1 are the ones that obtained the highest coordinates on this factor. Either considering Figs. 4 and 6 together or considering Fig. 5, we can explain the fact that systems in the right top corner perform specifically well on the topics that are in the same direction.

CA shows that T389 (out of the screen Fig. 4, very right part, see Fig. 5 too), T383, T397, and at a lower level T356, T376, T393, T372, and T394 for example are typical of the systems that contribute positively to factor 1. T379, T383 and at a lower level T397, T393, T386 are associated with systems that contribute positively to factor 2. Since these systems correspond to systems that perform well, those topics are "easy" for systems that are in the top right corner. Indeed, going back to the AP matrix, iit98ma1 is the best for T389 (same direction when considering Figs. 3, 5); this is a hard topic (average over

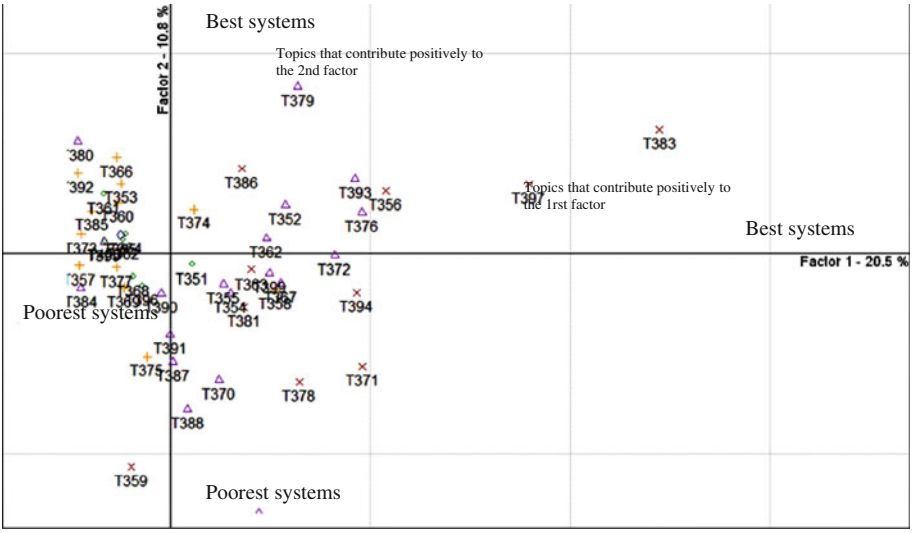

Fig. 4 2 First factors of the CA using AP matrix from TREC 7 ad hoc, topics only

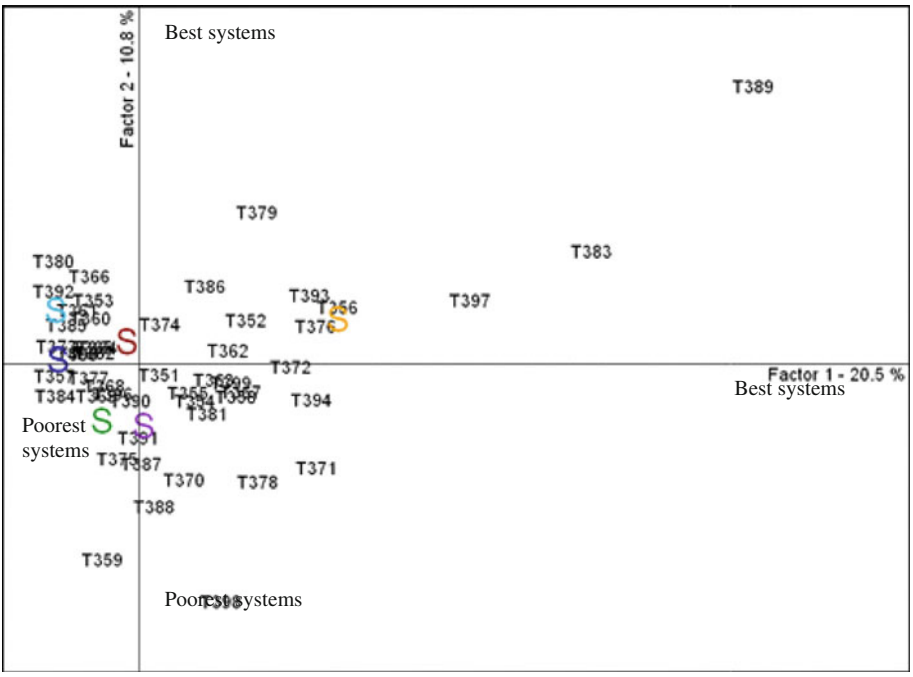

Fig. 5 2 First factors of the CA using AP matrix from TREC 7 ad hoc. Both topics and system centroids are displayed

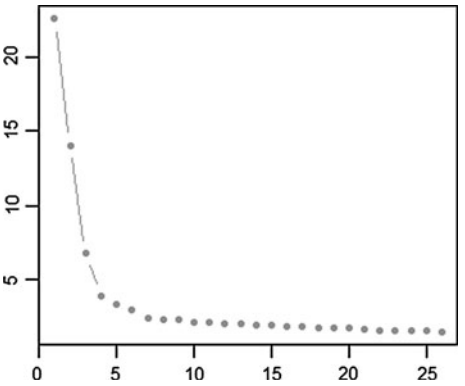

Fig. 6 An example of the distance between topic clusters

systems is 0.0250, iit98ma1 gets 0.4906. If one draws a vector from the origin of factors 1 and 2 to iit98ma1, we would see that T389 is in the same direction. T397 is also a hard topic (average AP over systems is 0.0980), and CLARIT08RANK performed best on it (0.6089). This is thus an atypical topic since on average systems perform poorly on it but a few systems succeeded. T383 is also a hard topic (average is 0.0593); the best systems do well on this topic as well. On the contrary, T376 is an easy topic (average 0.5179); LNaTitDesc7 is the system that performed the best on this topic, followed by most of the best systems. The same occurs for topic T372, the systems that rank the best for this topic are among the best systems. Still, considering factor 1 but on the opposite side (negative coordinates), T357 for example, is opposite to systems that perform well. This topic is not

that hard (0.1468), but the first system from the best system cluster appears at rank 15 and the second ranks 41! Best systems fail on this topic.

Topics that are close to the origin of the representation of factors are the ones that are typical (as opposed to atypical) meaning that they behave as other do. They cannot distinguish systems on the right side of factor 1 (which happen to be good systems) from systems that are on the left side of factor 1 (poor systems). Such topics are for example T351, T390, and T374. For these topics, some good systems fail, some bad systems succeed. For T390 for example (average AP over systems 0.0959), the system that performed the best is CLARIT98RANK (rank 1 for this topic), a system that belongs to the best system cluster (cluster 1 in Fig. 2). The next system that belongs to this cluster gets rank 13 for this topic. We can conclude that systems from the best system cluster lack a specific behavior for T390: this topic cannot distinguish this cluster from others.

Factor 3 (see "Appendix 4") allows one to distinguish cluster 2 of systems. When displaying topics at the same time, it is possible to see that T352, T390, T385, T376, and T357 are the topics that contribute the most in the same direction as systems from cluster 2 as opposed to T363 that contributes oppositely.

Figure 4 displays the combination of the results from CA and HC (see Sect. 4.2 for the reason behind combining both). In Fig. 4, a color (for digital version) and a specific plot shape have been associated with each cluster of topics detected in the first phase. For example the topics belonging to the cluster that appears laston the dendrogram in Fig. 1 are plotted in brown and crosses ($\times$) in Fig. 4 ("hardest" topics). In yellow and "+" are drawn the easy topics (third cluster). This makes it easy to observe that hard topics are mainly on the right side of factor 1 while easy topics are mainly on the left side.

We did the same type of analysis considering other measures. Baccini et al. (2011) show that P5 for example, even if it is correlated to MAP do not behave the same way. The clustering we obtained using P5 as well as the factorial analysis results show that system clusters can be observed in the same way as when MAP is used (even if the clusters differ of course in content).

4.4 Discussion

From the analysis we provided in the previous sections, we can conclude that some systems are better for processing some topic clusters, but we cannot conclude that they are better for processing all the topics of these clusters. We do notice that the hard topics are on the right side of factor 1, as are the best systems. This observation shows that the best systems are better for processing difficult topics (that could be a reason why they are considered as the best systems but not the only one since results are averaged over topics). Easy topics are on the opposite side of factor 1. The best systems are not necessarily the best systems to treat those topics.

Analyzing the two Figs. 4 and 6 simultaneously, we would be inclined to conclude that some specific topics should be treated by one selected system: T389 by iit98ma for example. Going a step further, we could develop a new system combination technique that learns which system or type of systems to use according to a given topic. This would be useful when queries are repeated in real systems: learning once for all the future occurrences of the same query. It could be somehow costly to do, but worth doing anyway if the repeated queries were frequent enough. It might be too costly if this has to be done on each query that occurs. Therefore, it would be useful to predict which queries have the highest probability of being repeated and concentrate the effort on these queries.

A motivation of defining a method that would learn the system to use for a given query is based on the fact that in real systems queries are repeated over time. Various studies

validate this hypothesis showing a significant proportion of repeated queries: 15% (Smyth et al. 2004), 17% Tyler and Teevan (2010), 33% (Teevan et al. 2007) and a little over 50% (Sanderson and Dumais 2007). In addition, Zhang and Lu (2009) provided features that help in predicting which query is the most likely to be repeated.

From these studies, it seems reasonable to promote an approach that would learn which method or system or group of systems would be the best to process a given query, given the fact that the cost would be for some of the queries only, the ones that are predicted to be repeated. Our method aims at proposing such a method that learns which system is the best.

In the next section, we present a new method to combine systems. This method takes advantage of the analysis made in Sect. 5 and is based on system selection on a per topic basis.

5 Fusing systems according to our findings

The analysis like the one we reported in the previous section leads us to propose a new type of combination method. We considered the following hypothesis: systems should be selected on the basis of their non-correlation. More precisely, our hypothesis is that considering complementary systems in terms of dependency as defined by clustering should be more effective than combining similar systems (systems that belong to the same cluster). In our approach, we consider two systems as complementary if they are not effective for the same topics, that is, if they do not belong to the same cluster as detected by the analysis. This perspective differs from other related works that rather concentrate on overlapping of the retrieved document lists (Beitzel et al. 2003) (Croft 2000).

5.1 Methods

We propose several variants of the same approach. For all of them, the idea is to combine systems not by aggregating the retrieved document lists, but rather by selecting one of the retrieved document lists.

The first variant is the most natural: with each topic is associated a single system. We call this method OneT2OneS: it is supposed to generate the best results in terms of effectiveness since it optimizes the performance measure of each individual topic. Using this method, a non-desired effect could be that a system succeeds by chance on a topic; it will be hazardous to propose a meta system that can exploit this in a general case. To eliminate this effect, we consider the OneT2ClusterS. In this variant, with each topic, the method associates the system cluster that should process it. From a cluster one representative system is chosen. This method is expected to perform less well than OneT2OneS but to be more robust. Another reason to introduce this variant is that it reduces the number of systems to use. Once clustered and a representative chosen, there is a limited number of systems involved. The last variant is the ClusterT2ClusterS. It aims at studying the behavior of the approach when topic clusters are considered.

5.1.1 OneT2OneS (one topic to one system)

The main goal of the method OneT2OneS is to associate with each topic the system that should be used to process it (one topic, one system). Our method differs from learning to rank methods (Liu et al. 2010) in which the best ranking method is not topic dependent. On the contrary, we propose an approach that selects the system according to the topic.

More precisely, for each topic, we select the system that maximizes the performance measure that we consider (e.g. AP). To evaluate the method, we consider the full set of topics. We learn the best system to use for each topic on a training document set and evaluate the results on the same topics, but on a test document set that differs from the training set.

5.1.2 OneT2ClusterS (one topic to one system cluster)

Rather than considering the best system for each topic, we calculate the best system cluster for each topic. For this, we first cluster the systems (see Sect. 4, hierarchical clustering using the Ward criterium + K-means). We define the representative system for each cluster as the system that obtains the best value of the measure for this cluster (say max(AP) for example). When two clusters "win", the ultimate winner is the system that gets the best value of the measure over the topics. Compared to OneT2OneS, this method aims at eliminating some systems that may have very unusual behavior, namely, being very bad on all topics except one or a few.

Again, to evaluate the method, we learn first on the training document set and test on the test document set, all the topics are considered both for training and testing.

5.1.3 ClusterT2ClusterS (one topic cluster to one system cluster)

In the ClusterT2ClusterS method, we analyze if the results are topic cluster dependent. For example, we want to know if the method performs better on hard topics or better on easy topics, or if there is no evidence that topic difficulty has an impact on the effectiveness of the method.

5.2 Evaluation

As noted in the previous section, a learning phase is needed to evaluate our method.

Our method is topic dependent. For this reason, the same topics have to be used in the training and the testing phase. On the other hand, documents on which the topics are processed should differ. In that way we will demonstrate that it is possible to learn our method on a subset of documents and that the function learned can be applied successfully on other sets of documents. The document collection is thus split into two partitions: a training document set and a testing document set.

To evaluate our method we applied tenfold cross validation.

For this, we consider the TREC ad hoc collection and make a partition: all the topics are used in the training phase, but the documents which are involved are partitioned according to a learning part (2/3 of the total) and a testing part (1/3). Rather than considering each document as independent, we group them together according to the beginning of their identifiers (e.g. all the documents that begin with `FR940105` will be either part of the training or of the testing set). In TREC 7, 2/3 of these document groups belong to the training, the remaining 1/3 to the testing set. Runs and qrels are processed in order to take into account this split. More precisely, each run is split into training and testing runs: each document from the current run goes to the training run part if the document belongs to the training document set; to the testing run otherwise. The qrels files are treated the same way. A single partitioning of the data in this way, however, is not enough to make solid conclusions since there is still an element of chance. For that reason, we performed 10 different partitions of the collection. For each partition, documents fall into training or

testing sets following a uniform random function. We learn the best system or best system cluster to use on each training document set. The learning consists in maximizing the MAP (considering the retrieved document list and qrels). After the training phase, we use the learned system or system cluster on the corresponding training document set. Therefore there are 10 experiments—labeled exp1,…, exp10 and corresponding results. We then averaged the results over the 10 experiments.

In the following tables, the value of MAP for our method resulting from the system combination method is followed by a "*" when the difference compared to the best system is statistically significant. Following the recommendations of Smucker et al. (2007) and Hull (1993), we test the statistical significance of the difference in average between two approaches using the Student's paired t test. The difference is computed between paired values corresponding system scores for all the topics and the difference between the tested samples is said to be statistically significant when $p < 0.05$. Although this test theoretically requires a normal data distribution it is robust to violations of this condition (Hull 1993).

5.2.1 OneT2OneS: training and testing phases on MAP

In Table 3, we present the detailed results using the tenfold training collections.

The baseline corresponds to the best system without considering any training; this is the best system that officially participates in TREC 7 (second row). It is not surprising that OneT2OneS (third row) outperforms the best system on training data, since the principle of the training is to select the best system for each topic (the best system on average is not necessarily the best for all the topics).

When considering the testing collections (Table 4), OneT2OneS still very much outperforms the best system.

Learning the best system to use for a topic improves MAP by about 21% on average.

5.2.2 OneT2ClusterS: training and testing phases on MAP

In Table 5, we present the detailed results we obtained using the tenfold testing collections (training is not presented here). In the second row, we indicate MAP of the best system that participates in TREC 7. The third row (OneT2ClusterS_30) corresponds to the resulted obtained for our system when the number of system clusters is set to 30 whatever the subcollection (1/3 of the total number of systems). In the fourth row (OneT2ClusterS_*), the number of system clusters is not set beforehand. We consider rather the distance in between the system clusters in the training phase and stop when it drops drastically. The number of system clusters resulting from this process is between 9 and 15, depending on the collections. The fifth row indicates this number of system clusters that have been defined in the training phase following the process we just explained.

On average, OneT2ClusterS improves MAP, either considering a relatively large number of clusters or considering a smaller number of clusters that depends more on the

Table 3 MAP when using OneT2OneS on the training phase on TREC 7 ad hoc

Experiment	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp10	Average
Best system	0.381	0.3691	0.381	0.392	0.378	0.381	0.383	0.375	0.387	0.384	0.381
OneT2OneS	0.544*	0.533*	0.557*	0.565*	0.540*	0.544*	0.555*	0.540*	0.541*	0.597*	0.552 (+44%)

* Indicates a significant difference ($p < 0.05$ applying the Student's paired t test)

Table 4 MAP when using OneT2OneS on the testing phase on TREC 7 ad hoc (training on MAP)

Experiment	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp10	Average
Best system	0.387	0.416	0.391	0.393	0.417	0.397	0.379	0.405	0.414	0.384	0.398<
OneT2OneS	0.468*	0.521*	0.464	0.443	0.505*	0.495*	0.474*	0.489*	0.504*	0.445	0.481 (+21%)

structure of the hierarchical clustering. In the former case, MAP is improved by 20% over the best system and by 15% in the latter case. Despite the relative importance of the improvement, the difference is not statistically significant ($p > 0.05$ applying the Student's paired t test) for all the sub collections. However, it is important to note that in all the cases the method improves MAP.

We then consider the ClusterT2ClusterS method in order to have a deeper examination of the topic clusters.

5.2.3 ClusterT2ClusterS

In this section, we reconsider the results from the OneT2ClusterS method and express them with regard to the topic clusters. As a rough clustering, we consider in this section a 3 cluster partitioning of topics where they can be viewed as easy, hard, and average topics. We made this choice because the ideal number of topic clusters may differ from one collection to another.

For example, given the distance between topic clusters as presented in Fig. 6, a first relevant cutting level would lead to 3 clusters, then the next relevant pruning would be at 4 clusters; then the distance becomes smaller until a cutting that would lead to 7 clusters. Choosing 3 clusters makes sense, as does 4. Indeed, for any of the 10 collections, 3 or 4 clusters make sense with regard to the distance between clusters.

Table 6 indicates the results, with each topic cluster presented separately. In each group of two rows, the first one corresponds to the best system that participates in TREC 7, considering the topics from the studied topic cluster. The second row corresponds to the results we obtained using our method on the same selected topics. The best improvements are not obtained on hard or easy topics. MAP improves most on the topics in-between (+24%, statistically significant)

5.3 Further discussions

Our results are difficult to compare to published results specifically in the fields of data fusion and learning to rank methods.

– CombMNZ-like functions consider two different document lists that they fuse before evaluating the resulting fused lists. The functions we named the CombMNZ-like functions basically differ in the way the documents are fused (considering document ranks or document scores, giving more weight to documents that are retrieved in the two lists, etc.). Fusing techniques can be applied to more than two lists, even to the full set of ranked lists that are available. We thus could have compared our meta system to the CombMNZ of the systems. However, a major difference with our approach is that no training phase is used for CombMNZ. A training phase could be useful if one aims at learning which systems to fuse. To our knowledge there is no publication that reports such method and results. However to give an idea, if we fuse any two runs from TREC

Table 5 MAP when using OneT2ClusterS on testing phase on TREC 7 ad hoc

Experiment	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp 10	Average
Best system	0.387	0.416	0.391	0.393	0.417	0.397	0.379	0.405	0.414	0.384	0.398
OneT2ClusterS_30	0.460	0.513	0.474	0.445	0.469	0.491*	0.486*	0.498*	0.501*	0.448	0.478 (+20%)
OneT2ClusterS_*	0.454	0.503*	0.463	0.436	0.479	0.458	0.455*	0.470	0.446	0.445	0.461 (+15%)
Number of clusters	11	14	10	11	9	11	12	12	13	15	

Table 6 MAP when using ClusterT2ClusterS on testing phase on TREC 7 ad hoc

Experiment	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp 10	Average
Best on hardest topics	0.349	0.378	0.297	0.315	0.324	0.333	0.303	0.348	0.337	0.320	0.330
ClusterT2ClusterS on hardest topics	0.364	0.429*	0.364*	0.329	0.349	0.312	0.386*	0.390	0.353	0.323	0.360 (+9%)
Best on easiest topics	0.592	0.578	0.618	0.511	0.740	0.560	0.631	0.779	0.697	0.571	0.628 (+11%)
ClusterT2ClusterS on easiest topics	0.686	0.668	0.679	0.664*	0.844	0.644	0.599	0.800	0.712	0.676*	0.695
Best on average	0.370	0.403	0.470	0.469	0.445	0.397	0.400	0.428	0.439	0.376	0.420
ClusterT2ClusterS on average topics	0.487*	0.526*	0.551*	0.522	0.540*	0.524*	0.501*	0.559*	0.497	0.479*	0.519 (+24%)

* Indicates statistically significant differences. % of improvement is calculated as compared to the best result considering the same group of topic difficulty (e.g. 24% is the improvement from 0.420 to 0.519)

Table 7 Summarization of the results obtained by the three methods we described (testing phase)

Method	MAP
Best system	0.398
OneT2OneS (test)	0.481 (+21%)
OneT2ClusterS (test)—30 clusters	0.478 (+20%)
OneT2ClusterS (test)—12 clusters (average)	0.461 (+15%)

7 using CombMNZ, and keep only the best fused pair, MAP is 0.49 (best single system 0.37). Notice that this can be seen as a training phase as it implies knowing which systems to fuse to obtain the best result. The meta system presented in this paper obtains 0.55 in the training phase (and 0.48 in the testing phase).

– In learning to rank (Cao et al. 2007), like in our approach, there is a training phase and a testing phase. However, in learning to rank, the task is to learn the ranking function, given details on topics and ideal ranked document lists. The ranking function is designed to be the same whatever the topic is. Our hypothesis is different since we design a method that is topic-dependent.

6 Conclusions

The long-term objective of our work is to build a system that would be based on several IR techniques in the various stages of an IR process and would choose the appropriate ones according to the query it receives. Question-answering systems have used the idea: depending on the type of query (who, what, etc.) the system receives; the method used to extract the answer (or the type of objects the system retrieves) differs. However, generalizing this principle to any type of query is not trivial.

This paper discusses some new directions towards this end. To do this, our first goal was to analyze system results on the same users' needs in new ways so that we could show that some new paths could be explored when considering system adaptation and/or system combination. We therefore carried out an in-depth analysis of some results obtained in the TREC environment.

When analyzing systems,

– We show that systems can be clustered according to the way they behave on topics, and topics can be clustered according to the way systems perform: systems that perform similarly on the same topics will be clustered together and vice versa. We observed that the best systems cluster together and that the poor systems cluster together too. The same occurs for topics: the easier topics cluster together and the hardest topic cluster together. Other system clusters are less related to system effectiveness whereas topic clusters are.
– We confirm the hypothesis that the most closely correlated systems are different versions of the same system. The variability in their performance according to topics is small compared to the variability between different systems. Indeed, generally speaking, TREC participants submit several runs (same task, same year), and those runs do not differ strongly. This could suggest that rather than trying to tune some parameters, we should consider radically different techniques to treat some types of topics.

– Using CA, we show that some topics are highly correlated with the best systems meaning that the systems behave differently for those topics than the other systems. Again considering the principal factors, we show that some successful systems are orthogonal in a way: they do not succeed for the same reason; that is to say not because of the same topics.

Based on these findings, we propose a new way of combining systems, in a topic-based way.

– According to literature in the domain of data fusion (Croft 2000; Wu and McClean 2006), we made an assumption that systems to be combined should be independent. However, contrary to the usual fusion techniques, independence is not based on the lists of ranked documents, nor on the overlapping of relevant and non-relevant documents, but on the fact that they perform differently on a subset of topics (system A is good for topic 1 and bad for topic 2 and system B gets opposite results). We promote a way to combine complementary systems, depending on their profiles based on their effectiveness on topics on a training document set.
– We present three versions of the same method. The three methods can be seen as a single method with different parameters. The method makes use of system clustering (hierarchical clustering + K-means on these clusters). For each system cluster, the representative system which will process the topics that fall into its scope is calculated. For each topic, we learn what the best system cluster is and thus what system will be used to process it. In turn, topics are clustered (hierarchical clustering + K-means); we can then see how much the method improves the results on each topic cluster. Table 7 summarizes the results. In the OneT2OneS method, we consider as many system clusters as systems (one system per cluster) and a single topic cluster. In the OneT2ClusterS method, there are less system clusters than systems and there is a single topic cluster. Finally, in the ClusterT2ClusterS method, there are fewer system clusters than systems (as in the OneT2ClusterS) but more than one topic per cluster.

Evaluation shows that:

– MAP can be improved by 20% after a training phase,
– Considering a smaller number of system clusters does not dramatically decrease performance,
– Improvement is better on topics of average difficulty, neither the hardest (as one would have intuitively thought), nor the easiest.

Though this method is only applicable for now to queries that are known in advance, nevertheless it is worth using it when queries are repeated. Several studies (Smyth et al. 2004; Sanderson and Dumais 2007; Teevan et al. 2007; Tyler and Teevan 2010) demonstrate that repetition occurs frequently in real applications. In addition, some studies have shown that repetition in some cases is predictable (Zhang and Lu 2009). For these reasons, the fact that the queries on which learning should be applied is not a limitation of the work. In addition, in our experimental framework, we considered relevance judgment in the learning phase; in real world applications, it is possible to consider users' actions to induce a relevant document set.

Future work will be conducted work on the systems that should be used in the method. Indeed, the analysis presented in Sect. 4 shows that at least considering TREC runs, variants of a same system lead to comparable effectiveness and that the method should pick up different systems. In future work, we will consider system characteristics (indexing

method used, query reformulation method, ranking function used, system parameters, etc.). Then we will analyze in depth the documents specific parameters setting retrieves in order to know if we can extract patterns in the way systems perform and correlate this with system clusters and methods or models used in systems. Complementarily, we will investigate query formulation in depth. Considering TREC for example, most participants consider two parts of the topic description (the title and the description of the information need) in order to create the queries to be sent to their systems. These queries have different characteristics, for example their syntactic features (Mothe and Tanguy 2005). To what extent the topic features are correlated with system performance is one of the new perspectives we are studying.

Acknowledgments We thank Fionn Murtagh, University of London, for his valuable advices and the very interesting discussion on CA and PCA usage and properties. We also thank Alain Baccini and Sébastien Déjean, from the Institut Mathématique de Toulouse. This work was supported in part by the ANR Agence Nationale de la Recherche, through the CAAS project and FREMIT federation.

Appendix 1

See Table 8.

Table 8 AP for TREC 7 ad hoc considering topic clusters

Easiest topics cluster 1 Blue	Easy topic cluster 2 Green	Topic cluster 3 Yellow	Hard topics cluster 4 Violet	Hardest topics cluster 5 Brown
MAP O.628	MAP 0.358	MAP 0.231	MAP 0.153	MAP 0.046
1-T365	4-T351	T353	T352	44-T356
	6-T361	T357	T354	45-T359
	3-T364	T358	T355	41-T363
	2-T368	T360	T362	47-T371
	5-T382	T366	T367	48-T378
	8-T396	T369	T370	46-T381
	7-T400	T373	T372	42-T383
		T374	T376	50-T386
		T375	T379	49-T389
		T377	T380	43-T394
		T385	T384	40-T397
		T392	T387	
			T388	
			T390	
			T391	
			T393	
			T395	
			T398	
			T399	

The easiest and most difficult topics are preceded by their rank in terms of difficulty (1 being the easiest topic 1-T365 and 50 the most difficult topic, 50-T386)

Appendix 2

See Table 9.

Table 9 AP for TREC 7 ad hoc considering system clusters

Sys_Orange 7 syst.	Sys_Rose 21 syst.	Sys_Violet 23 syst.	Sys_Dark_Blue 11 syst.	Sys_Green 15 syst.	Sys_Blue 26 syst.
CLARIT98CLUS	Cor7A1clt	Brkly25	KD70000	AntHoc01	APL985L
CLARIT98COMB	LNaTit7	Brkly26	KD71010q	fub98a	APL985LC
CLARIT98RANK	LNaTitDesc7	Cor7A2rrd	KD71010s	fub98b	APL985SC
iit98ma1	LNmanual7	Cor7A3rrf	ScaiTREC 7	gersh1	Brkly24
t7miti1	acsys7al	FLab7ad	dsir07a01	Ic98san3	ETHAB0
uwmt7a1	att98atc	INQ501	dsir07a02	Ic98san4	ETHAC0
uwmt7a2	att98atdc	INQ502	jalbse013	jalbse011	ETHAR0
	att98atde	INQ503	kslsV1	jalbse012	FLab7at
	bbn1	MerAdRbtnd	lanl981	nectitech	FLab7atE
	mds98t	MerTetAdtnd	nthu3	nsasgrp3	LIAClass
	mds98t2	acsys7mi	umd98a2	nsasgrp4	LIArel2
	mds98td	fsclt7m		nthu1	LIAshort2
	ok7am	harris1		umd98a1	MerAdRbtd
	ok7as	ibms98a		unc7aal1	acsys7as
	ok7ax	ibms98b		unc7aal2	fsclt7a
	pirc8Aa2	ibms98c			gersh2
	pirc8Ad	iit98au1			gersh3
	pirc8At	iowacuhk1			ibmg98a
	tno7exp1	iowacuhk2			ibmg98b
	uoftimgr	Nectitechall			ibmg98c
	uoftimgu	nectitechdes			iit98au2
		tno7cbm25			nthu2
		tno7tw4			nttdata7Al0
					nttdata7Al2
					nttdata7At1
					uwmt7a0
Avg MAP 0.345	Avg MAP 0.262	Avg MAP 0.238	Avg MAP 0.045	Avg MAP 0.144	Avg MAP 0.170

Appendix 3

In the panel below, systems have been ordered according to increasing MAP (averaged over topics). Then their names have been replaced by a number and a color that represents the cluster they belong to. On the extremes (clusters numbered 1 and 4), clusters and system effectiveness are highly correlated; for the other clusters, this correlation is weaker, even if cluster numbered 2 tends to be poorer than the one numbered 1; this latter tending to be lower than the cluster numbered 3 which in turn tends to be lower than the cluster numbered 5.

Appendix 4

See Fig. 7

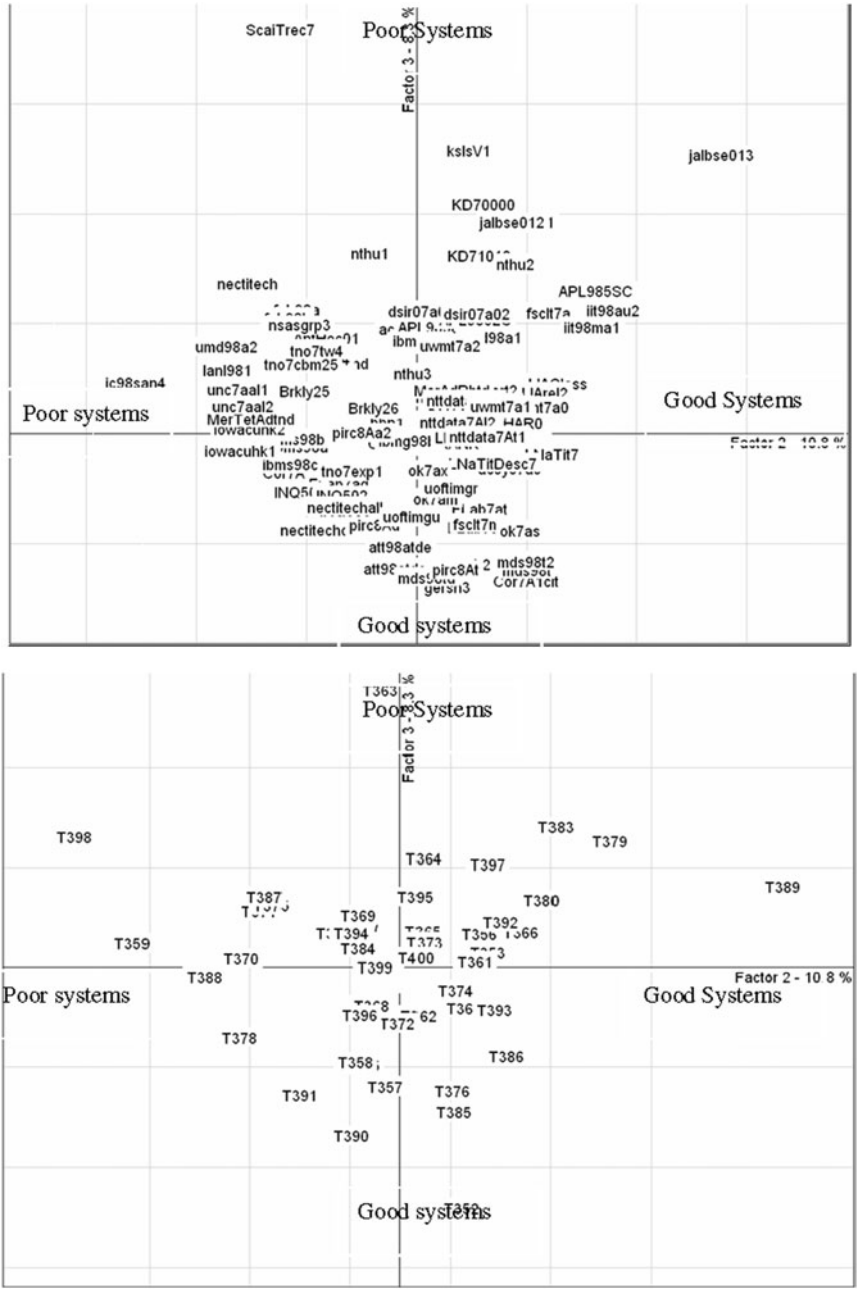

Fig. 7 Factors 2 and 3 of the CA using AP onTREC 7 ad hoc. The two factors correspond to 10.8 and 8.3% of total inertia. Systems are displayed in the *top figure* and topics on the *bottom figure*

References

Allan, J., Callan, J., Sanderson, M., Xu, J., & Wegmann, S. (1998). INQUERY and TREC-7. *The Seventh Text REtrieval Conference, NIST special publication 500-242* (pp. 201-216).

Aslam, J. A., & Montague, M. (2001). Models for metasearch. *24th international ACM SIGIR conference on research and development in information retrieval* (pp. 276–284).

Avila, F., & Myers, D. E. (1991). Correspondence analysis applied to environmental data sets: A study of Chautauqua Lake sediments. *Chemometrics and Intelligent Laboratory Systems, 11*(3), 229–249.

Baccini, A., Déjean, S., Lafage, L., & Mothe, J. (2011). How many performance measures to evaluate information retrieval systems? *Knowledge and Information Systems*. doi:10.1007/s10115-011-0391-7, 2011.

Banks, D., Over, P., & Zhang, N.-F. (1999). Blind men and elephants: Six approaches to TREC data. *Information Retrieval, 1*(1–2), 7–34.

Beitzel, S. M., Frieder, O., Jensen, E. C., Grossman, D., Chowdhury, A., & Goharian, N. (2003). Disproving the fusion hypothesis: an analysis of data fusion via effective information retrieval strategies. *ACM symposium on applied computing SAC* (pp. 823–827).

Benzécri, J.-P. (1973). *L'Analyse des Données [data analysis], Vol. II. L'Analyse des Correspondances [correspondence analysis]*. Paris, France: Dunod.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. *International Conference on Machine Learning*, 129–136.

Carmel, D., Yom-Tov, E., Darlow, A., & Pelleg, D. (2006). What makes a query difficult? *International ACM SIGIR conference on research and development in information retrieval* (pp. 390–397).

Chrisment, C., Dkaki, T., Mothe, J., Poulain, S., & Tanguy, L. (2005). Recherche d'information: Analyse des résultats de différents systèmes réalisant la même tâche [*Information retrieval: analysis of different systems doing the same task*]. *Ingénierie des Systèmes d'Information, 10*(1), 33–57.

Croft, W. B. (2000). Combining approaches to information retrieval. In *Advances in information retrieval: Recent research from the Center for Intelligent Information Retrieval, Chap. 1*. Kluwer.

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting query performance. *International ACM SIGIR conference on research and development in information retrieval* (pp. 299–306).

Dkaki, T., & Mothe, J. (2004). Combining positive and negative query feedback in passage retrieval. *RIAO, 2004*, 661–672.

Elovici, Y., Shapira, B., & Kantor, P. B. (2006). A decision theoretic approach to combining information filters: An analytical and empirical evaluation. *Journal of American Society for Information Science, 57*(3), 306–320.

Evans, D., & Lefferts, R. (1994). Design and evaluation of the clarit-trec-2 system. *2nd Text REtrieval Conference (TREC-2), NIST special publication 500-215* (pp. 137–150).

Ferraretti, D., Gamberoni, G., & Lamma, E. (2009). Automatic cluster selection using index driven search strategy, 11th international conference of the Italian Association for Artificial Intelligence Reggio Emilia on Emergent Perspectives in Artificial Intelligence (AI*IA '09:). *Lecture Notes in Computer Science, 5883*(2009), 172–181.

Fox, E. A., & Shaw, J. A. (1994). Combination of multiple searches. *2nd Text REtrieval Conference (TREC-2), NIST special publication 500-215* (pp. 243–252).

Greenacre, M. J., & Blasius, J. (Eds.). (1994). *Correspondence analysis in the social sciences: Recent developments and applications*. Academic Press. ISBN: 0-12-104570-6.

Harman, D. (2000). What we have learned and have not learned from TREC. *The BCS/IRSG 22nd annual colloquium on information retrieval research, Cambridge* (pp. 2–21).

Harman, D., & Buckley, C. (2004). The NRRC reliable information access (RIA) workshop. *International ACM SIGIR conference on research and development in information retrieval* (pp. 528–529).

Harman, D., & Buckley, C. (2009). Overview of the reliable information access workshop. *Information Retrieval, 12*(6), 615–641.

He, B., & Ounis, I. (2003). University of Glasgow at the robust track—a query-based model selection approach for poorly-performing queries. *The twelfth text retrieval conference, SP 500-255* (pp. 636–645).

Hubert, G., & Mothe, J. (2007). Relevance feedback as an indicator to select the best search engine—evaluation on TREC data. In *Proceedings of the tenth international conference on enterprise information systems, Vol. ISAS—Information Systems Analysis and Specification (ICEIS 2007)* (pp. 149–154).

Hubert, G., & Mothe, J. (2009). An adaptable search engine for multimodal information retrieval. *Journal of American Society for Information Science and Technology, 60*(8), 1625–1634.

Hubert, G., Mothe, J., & Englmeier, K. (2007). Tuning search engine to fit XML retrieval scenario. *Conference on Web Information Systems and Technologies, WebIST* (pp. 228–233). INSTICC Press.

Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. *International ACM SIGIR conference on research and development in information retrieval* (pp. 329–338).

Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Berlin: Springer.

Kamps, J., Geva, S., Peters, C., Sakai, T., Trotman, A., & Voorhees, E. (2009). Report on the SIGIR 2009 workshop on the future of IR evaluation. *ACM SIGIR Forum, 43*(2), 13–23. http://www.sigir.org/forum/2009D/sigirwksp/2009d_sigirforum_kamps.pdf.

Kurland, O. (2009). Re-ranking search results using language models of query-specific clusters. *Information Retrieval Journal, 12*(4), 437–460.

Lebart, L., Piron, M., & Morineau, A. (2006). Statistique exploratoire multidimensionnelle:Visualisations et inférences en fouille de données [*multidimensionnal exploratory statistics: Visualization and inferences in data mining*] (4th ed.). Dunod.

Lebart, L., Salem, A., & Berry, L. (1998). *Exploring textual data*. Dordrecht, Boston: Kluwer.

Lee, J. (1997). Analysis of multiple evidence combination. *International ACM SIGIR conference on research and development in information retrieval* (pp. 267–276).

Liu, B. (2006). *Web data mining—exploring hyperlinks, contents and usage data*. Berlin: Springer.

Liu, T.-Y., Joachims, T., & Zhai, C. (2010). Introduction to special issue on learning to rank for information retrieval. *Information Retrieval Journal, 13*(3), 197–200.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Symposium on mathematical statistics and probability, Berkeley* (pp. 281–297).

Madeira, S. C., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1*, 24–45.

Mandl, T., & Womser-Hacker, C. (2003). Linguistic and statistical analysis of the CLEF topics, advances in cross-language information retrieval. *Lecture Notes in Computer Science, 2785/2003*, 505–511.

Mizzaro, S., & Robertson, S. (2007). Hits hits TREC: Exploring IR evaluation results with network analysis. *International ACM SIGIR conference on research and development in information retrieval* (pp. 479–486).

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* (pp. 7–10).

Mothe, J., & Tanguy, L. (2008). Linguistic analysis of users' queries: Towards an adaptive information retrieval system. In *Proceedings of the international conference on signal image technologies and internet based systems (SITIS 2007)* (pp. 77–84).

Murtagh, F. (2005). *Correspondence analysis and data coding with Java and R, computer science and data analysis series*. Chapman & Hall/CRC. ISBN: 1-58488-528-9.

Murtagh, F., Starck, J.-L., & Berry, M. (2000). Overcoming the curse of dimensionality in clustering by means of the wavelet transform. *The Computer Journal, 43*, 107–120.

Ng, K. B., & Kantor, P. B. (2000). Predicting the effectiveness of nave data fusion on the basis of system characteristics. *Journal of American Society for Information Science, 51*, 1177–1189.

Robertson, S. (2009). Richer theories, richer experiments, the future of IR evaluation workshop. *International ACM SIGIR conference on research and development in information retrieval* (p. 4).

Robertson, S., & Walker, S. (1994). Some simple approximations to the 2-Poisson model for probabilistic weighted retrieval. *International ACM SIGIR conference on research and development in information retrieval* (pp. 232–241).

Rowe, B. R., Wood, D. W., Link, A. N., & Simoni, D. A. (2010). *Economic impact assessment of NIST's Text REtrieval Conference (TREC) program*. RTI project number 0211875.

Sanderson, M., & Dumais, S. (2007). Examining repetition in user search behaviour. *European conference on IR research, advances in information retrieval* (pp. 597–604).

Seber, G. A. F. (1984). *Multivariate observations*. Hoboken: Wiley.

Smucker, M. D., Allan, J., & Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. *ACM conference on information and knowledge management* (pp. 623–632).

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., & Boydell, O. (2004). Exploiting query repetition and regularity in an adaptive community-based web search engine. User model. User-adapt. *Interact, 14*(5), 383–423.

Teevan, J., Adar, E., Jones, R., & Potts, M. A. (2007). Information re-retrieval: Repeat queries in Yahoo's logs. *International ACM SIGIR conference on research and development in information retrieval* (pp. 151–158).

Tekaia, F., Yeramian, E., & Dujon, B. (2002). Amino acid composition of genomes, lifestyles of organisms, and evolutionary trends: A global picture with correspondence analysis. *Gene, 297*(1–2), 51–60.

Trotman, A. (2005). Learning to rank. *Information Retrieval Journal, 8*(3), 359–381.

Tyler, S. K., & Teevan, J. (2010). Large scale query log analysis of re-finding. *Third ACM international conference on web search and data mining (WSDM'10)* (pp. 191–200).

Voorhees, E. M. (2007). Overview of the TREC 2006. *The fifteenth Text REtrieval Conference (TREC 2006), NIST Special Publication 500-272* (pp. 1–16).

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association, 58*, 236–244.

Wen, J.-R., Nie, J.-Y., & Zhang, H.-J. (2002). Query clustering using user logs. *ACM Transactions on Information Systems (TOIS), 20*(1), 59–81.

Wilkins, P., Adamek, T., O'Connor, N. E., & Smeaton, A. (2006). Using score distributions for query-time fusion in multimedia retrieval. *8th ACM international workshop on Multimedia Information Retrieval (MIR'06)* (pp. 51–60).

Wu, S., & McClean, S. (2006). Improving high accuracy retrieval by eliminating the uneven correlation effect in data fusion. *Journal of the American Society for Information Science and Technology, 57*(10), 1962–1973.

Zhang, D., & Lu, J. (2009). What queries are likely to recur in web search? *International ACM SIGIR conference on research and development in information retrieval* (pp. 827–828).