Automatic Classification of Queries by Expected Retrieval Performance

Jens GRIVOLLA
jens.grivolla@univ-avignon.fr

Pierre JOURLIN
pierre.jourlin@univ-avignon.fr

Renato DE MORI
renato.demori@univ-avignon.fr

ABSTRACT

This paper presents a method for automatically predicting a degree of average relevance of a retrieved document set returned by a retrieval system in response to a query.

For a given retrieval system and document collection, prediction is conceived as query classification. Two classes of queries have been defined: *easy* and *hard*. The split point between those two classes is the median value of the average precision over the query collection. This paper proposes several classifiers that select useful features among a set of candidates and use them to predict the class of a query.

Classifiers are trained on the results of the systems involved in the TREC 8 campaign. Due to the limited number of available queries, training and test are performed with the leave-one-out and 10-fold cross-validation methods. Two types of classifiers, namely decision trees and support vector machines provide particularly interesting results for a number of systems. A fairly high classification accuracy is obtained using the TREC 8 data (more than 80% of correct prediction in some settings).

1. INTRODUCTION

In document retrieval applications it can be necessary to have a measure of confidence in the results found by the retrieval system. A good relevance estimation can serve as a criterion for a large choice of search strategies, e.g. selecting the "best" source of information when multiple document collections are available, applying additional expansion techniques according to the specific needs, enabling user interaction with the aim of improving the formulation of the request, etc.

These strategies can, in particular, include computationally expensive techniques that can be avoided if the result of information retrieval is predicted to be sufficiently good. One hint about the importance of having a specific treatment for particularly difficult queries is given by the recent introduction of a "robust" task ([Voorhees, 2003]) in the TREC (*Text REtrieval Conference*[1]) evaluation campaigns, organized by the NIST (*National Institute for Standards and Technology*).

The problem of finding a measure of confidence about the result of a retrieval system is a very difficult one, especially if we consider the rarity of available, high quality, human assessments. However, this problem seems to be addressed by a increasing number of researchers as will be shown in section 2 where related work is discussed. Motivations and description of our approach are introduced in section 3, details are provided in sections 4 and 5, while experiments are described in section 6.

2. RELATED WORK

So far, relatively few groups have conducted in-depth work on this performance prediction, and usually with rather moderate success. For TREC 6, it is noted that short queries (titles only) containing well-chosen keywords can give good results, whereas absence of those terms (queries without the titles) deteriorates the results [Voorhees and Harman, 1997]. It is suggested that the retrieval strategy should be adapted to the query length.

Great differences have been observed between short (title only) queries and the longer versions which yielded much better results on TREC 5 [Voorhees and Harman, 1996]. However, this difference is practically nonexistent on TREC 7 and TREC 8, where the participating system perform almost equally well on title only queries as on the ones including a full description of the information need. Bank, Over and Zhang approach the problem of classifying queries with regard to the performance obtained by different retrieval systems using six statistical methods [Banks et al., 1998]. The results are not satisfying, as is stated in the article: "None of the work we have done using the six approaches discussed here has provided the sort of insights we were seeking [...]".

On TREC 8, Karen Spärck Jones notes a strong correlation between the IR systems' performances and the quality of information about the query, as well as "query difficulty" in a general, non-technical sense [Spärck-Jones, 1999]. Following TREC 8, Rorvig evaluates the difficulty of query sets from different years [Rorvig, 1999]. He establishes criteria allowing to make predictions on the difficulty of sets of queries. However, those measures do not enable us to predict individual queries' difficulties. Rorvig confirms that simple measures such as query length are not very useful to make predictions:

[1] `http://trec.nist.gov`

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

> "Factors that cannot describe query difficulty are: (1) topic components (concepts, narratives, etc.), (2) topic length, (3) and topic construction (creating topics without regard to existing documents vs. the contrary practice). Document uniqueness is the only quantitative measure so far offered. Indeed, topic hardness appears to rest in that zone of phenomena that many can mutually observe, but cannot describe in terms that would eventually permit control."

An article by Claude de Loupy and Patrice Bellot at LREC 2000 explores some parameters that can help in estimating the difficulty of individual queries [de Loupy and Bellot, 2000]. One of the most successful attempts appears to be [Cronen-Townsend et al., 2002], using a score describing the *clarity*, i.e. lack of ambiguity of a query. A different approach is described in [Sullivan, 2001], based on the similarity between queries.

Amati, Carpineto and Romano propose a different measure (called $INFO_{DFR}$) that correlates well with average precision [Amati et al., 2004], based on the query term distribution in the top-ranked documents, as well as a measure used to predict the effect of query expansion ($INFOQ$). A method for Query-Sensitive Tuning (QUEST) to adapt the document ranking according to the query type has been presented at TREC 12 ([Amitay et al., 2003]).

The most recent publication may be [He and Ounis, 2004], which presents several query performance predictors that can be computed prior to the retrieval process. Their correlation with average precision is evaluated on the TREC 7 and 8 collections.

3. MOTIVATIONS AND APPROACH

A major departure from previous approaches to performance prediction is the simultaneous use of a number of features in order to exploit the power of the combination of their values. Performance prediction is seen as a classification problem in which a set of performance features are used to assign a confidence label or value to a query. Once a set of potential features has been identified, machine learning algorithms are used to train classifiers for the prediction task. Among others, Classification Trees (CT) and Support Vector Machines (SVM) have been used.

These devices are trained with corpora of data. Suitable corpora are the results of public competitions. As outlined in section 2, interesting observations about system performance were reported using the data obtained from the yearly TREC campaigns. In practice, the data of TREC 8 are suitable for training the classifiers because they have been obtained with a large collection of documents (approximately 500,000) and queries (50/year for the *ad-hoc*-track).

While TREC 8 is composed of different *tracks*, such as question answering, spoken document retrieval, and others, the experiments described in this paper are based on the data from the *ad-hoc* task, the non-interactive retrieval of documents for natural language queries formulated in English. The queries are available in several versions, consisting of a title (few selected key words), an additional description, and possibly a longer narrative describing in detail what information is needed. The submissions of several teams that participated in the TREC 8 campaign (see [Voorhees and Harman, 1999]) have been used for our experiments.

In order to conceive a method for predicting system performance for a given query about the content of a given archive, a set of reliability features has been created. Some features are computed on the basis of the query itself, depending on the information available, the retrieval results and possibly knowledge about the process having led to the results (ranking scores, etc.).

Using a corpus of queries for which the system retrieval performance was known, automatic classification techniques were applied in order to obtain predictions on the difficulty of the query. Machine learning techniques require at least two separate sets of data, one for training and another one for testing. Due to the limited number of available queries in the corpus, leave-one-out and 10-fold cross-validation methods were used. Some experiments were also performed using data from TREC 8 for training and data from TREC 7 for testing. Two types of classifiers were used to predict whether the query was easy (high precision expected on retrieved results) or hard (expected low precision). A set of classifiers were trained for each system. Feature selection was also performed by the training process.

4. FEATURES

A number of attributes describing retrieval situations has been investigated to form a set of candidate features. These attributes are extracted from any available information such as query terms, documents retrieved by one or several systems, etc. An empirical analysis of the easiest and hardest queries shows that there is a large variety of very different queries in both groups. Furthermore, it appears that no single attribute allows a clear differentiation between those two classes. A detailed description and discussion about the features considered would be too long. Thus, feature types are briefly introduced in the following, together with some details on representative features.

4.1 Empirical features

A number of features describing the query itself have been considered, such as the query length and different measures of ambiguity or specificity of the query terms (e.g. synonymy, number of senses, hyponymy). Confirming the results mentioned in [Rorvig, 1999, Spärck-Jones, 1999, Voorhees and Harman, 1997], no strong correlation between query length and retrieval performance has been observed. Some of the other measures show a relation with the average precision obtained for the query.

4.2 Entropy and Pairwise Similarity

Taking into account the results from an initial retrieval run one can add more sophisticated measures that depend on the set of documents. One such measure is the entropy of the set of the K top-ranked documents for a query. It can be expected that good retrieval results will provide a more homogenous set of documents. Therefore, the entropy of the retrieved document set (or part of it) should be higher when the performance achieved for a given query is bad.

The set of the K top-ranked retrieved documents can be seen as a source of information. The entropy of this source can be computed using a statistical language model (LM) as proposed in [Carpineto et al., 2001]. If the entropy of the set is high, the linguistic structure of the documents is highly variable. We assume that a large linguistic variability is cor-

related with a higher risk that some retrieved documents are not relevant. The probability of retrieving non relevant documents increases with K. Since a single, long, non relevant document may cause a significant increase in entropy, it is advisable to keep K small. Taking these observations into account, only unigram probabilities can be estimated with acceptable accuracy for the LM and the entropy is defined as:

$$H = -\sum_{w \in W} P(w) \cdot \log P(w)$$

where W is a lexicon of keywords, and $P(w)$ is the probability of the word w in the document set. $P(w)$ is estimated as follows:

$$P(w) = \frac{\sum_{d \in D} N_d(w) + \epsilon}{\sum_{v \in W} \sum_{d \in D} N_d(v) + |W|\epsilon}$$

where D is the set of documents on which to calculate the entropy, $N_d(w)$ the number of occurrences of w in d, and ϵ is a constant used to overcome the well-known zero-frequency estimation problem. For a first test we chose $W = V$, V being the complete term-vocabulary of the document collection, i.e. without stop words, and after applying the Porter stemming algorithm. Correlation between entropy and retrieval performance is relatively weak.

By limiting the vocabulary to the most frequent words from each document instead of calculating the entropy over the entire vocabulary, a better correlation was observed compared to using the complete vocabulary.

A score of the same type as the entropy is the mean cosine similarity of the documents (MCS). We calculated this in the usual manner, using the same stoplist and porter stemming as for our other measures, and the base form of $TF.IDF$ term weighting:

$$w_{ij} = tf_{ij} * \log \frac{|D|}{df_i}$$

(assigning the weight w_{ij} to term T_i in document D_j).

This measure, as well as entropy over a limited vocabulary, was computed for different values of K.

4.3 Retrieval scores

Among the best features are some of the scores provided by the different systems for ranking the retrieved documents related to a given query.

Most document retrieval systems rank the documents for a given query by a score that it more or less directly derived from either $TF.IDF$-weighted similarity between document and query unigram vectors (e.g. Smart [Salton, 1989]), bayesian probability ($P(d|q)$, typically based on term occurrence probabilities (e.g. Okapi [Robertson et al., 1996]) or bayesian inference networks (e.g. InQuery [Callan et al., 1992]).

These scores have been proven to be successful for the relative ranking of the documents in response to a query, they are not generally expected to be an absolute measure of relevance (although some of the retrieval models do in theory attempt to calculate the probability of relevance).

In a preliminary study, it was observed that the scores used by e.g. Okapi or InQuery were strongly correlated with the retrieval performance for the corresponding query. However, some other systems (such as the IBM system for TREC 8) use scores that were not very useful as retrieval quality predictors, even if they lead to a good mean average precision.

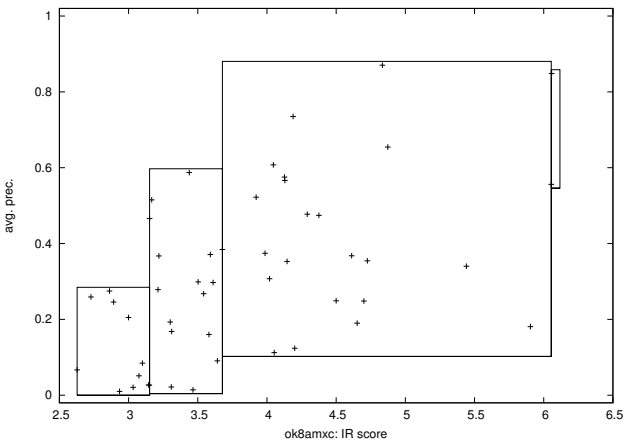

Figure 1: ok8amxc: 4-class split on retrieval score

We have used different transformations and normalizations of these scores. Based on the scores assigned by the document retrieval system to each document (for a given query), we have thus calculated scores for the queries themselves. A query can for example be represented by the mean score of the N top-ranked documents, the score of the n^{th} document, or the ratio of the scores assigned to the first and the n^{th} document.

Figure 1 shows the relation of ranking score to average precision for the Okapi system (ok8amxc), using the mean score over the top 20 documents. No query with a score below 3.15 reached an average precision above 0.28, and no query with a score under 3.7 achieved better average precision than 0.59. For several other TREC 8 participants, similar boundaries can be determined.

Comparing the results obtained for all systems having participated in TREC 8, we have noticed that the correlation of these measures based on the different retrieval scores with average precision achieved for a query ranges from very strong to inexistent, independently of the performance of the corresponding document retrieval system.

4.4 Summary

Table 1 shows the rank correlations ρ as well as p-values (on Spearman's rank correlation) of some of the features we have used with the average precision achieved by a small selection of systems. As has been said before, none of the features show a perfect correlation with query performance, but many of them can still be useful in automatically assessing query difficulty, especially when used in combination.

It is interesting to note that the correlations vary considerably between systems.

5. CLASSIFICATION

Having determined a set of attributes computable on the basis of the query itself and (depending on the information available) the retrieval results and possibly knowledge about the process having led to those results (ranking scores, etc.), we represent each query (or query-response entity) as a vector, the dimension of which depends on the chosen attributes.

	feature	ok8amxc		INQ603		ibms99a		tno8d4		mds08a2	
		rho	p-value	rho	p-value	rho	p-value	rho	p-value	rho	p-value
docs	entropy (10 docs)	-0.0950	5.19E-01	**-0.2423**	**9.35E-02**	**-0.2841**	**5.07E-02**	**-0.3223**	**3.13E-02**	**-0.3439**	**2.28E-02**
	entropy (15 docs)	**-0.2778**	**5.90E-02**	-0.0607	6.78E-01	-0.1851	2.18E-01	**-0.2562**	**9.33E-02**	-0.0421	7.88E-01
	mcs (50 docs)	0.1359	3.46E-01	**0.2514**	**7.83E-02**	0.0383	7.91E-01	0.1778	2.16E-01	-0.0151	9.17E-01
retrieval score	mean (10 docs)	**0.5194**	**1.39E-04**	**0.7580**	**< 2.2E-16**	0.1802	2.10E-01	**0.5017**	**2.48E-04**	**0.5409**	**6.55E-05**
	ratio (50th doc)	**-0.4464**	**1.29E-03**	**-0.5304**	**9.53E-05**	**-0.3082**	**2.99E-02**	**-0.3688**	**8.76E-03**	**-0.4638**	**7.90E-04**
	score (1st doc)	**0.5043**	**2.28E-04**	**0.5935**	**8.55E-06**	0.1870	1.93E-01	**0.4563**	**9.76E-04**	**0.5458**	**5.49E-05**
	score (5th doc)	**0.5223**	**1.26E-04**	**0.7554**	**< 2.2E-16**	0.1795	2.11E-01	**0.5029**	**2.39E-04**	**0.5148**	**1.62E-04**
query title	length	**-0.2749**	**5.37E-02**	-0.1032	4.75E-01	-0.2149	1.34E-01	-0.2258	1.15E-01	-0.1233	3.93E-01
	senses (de Loupy/Bellot)	0.0997	4.90E-01	0.2139	1.35E-01	0.1270	3.78E-01	0.1095	4.48E-01	0.0876	5.44E-01
	senses (total)	-0.1372	3.41E-01	-0.2029	1.57E-01	-0.1419	3.25E-01	**-0.2758**	**5.29E-02**	-0.1437	3.18E-01
	synonyms (avg.)	-0.1464	3.25E-01	**-0.2499**	**9.04E-02**	-0.1302	3.82E-01	-0.2257	1.27E-01	-0.2026	1.72E-01
	synonyms (total)	-0.1324	3.74E-01	-0.2036	1.70E-01	-0.1158	4.37E-01	**-0.2569**	**8.14E-02**	-0.2079	1.60E-01
query title + description	length	-0.1599	2.66E-01	-0.0415	7.74E-01	-0.0070	9.62E-01	-0.1622	2.60E-01	-0.0714	6.21E-01
	senses (de Loupy/Bellot)	0.0760	5.99E-01	0.1953	1.74E-01	0.1711	2.34E-01	0.0061	9.66E-01	-0.0075	9.59E-01
	senses (total)	-0.1607	2.64E-01	-0.2164	1.31E-01	-0.0784	5.88E-01	**-0.2435**	**8.85E-02**	**-0.2462**	**8.49E-02**
	synonyms (avg.)	-0.0207	8.86E-01	-0.1544	2.84E-01	-0.0259	8.58E-01	-0.1272	3.78E-01	-0.1205	4.04E-01
	synonyms (total)	-0.1095	4.48E-01	-0.1637	2.55E-01	0.0078	9.57E-01	-0.1788	2.14E-01	-0.1824	2.04E-01

Table 1: features and their correlation with average precision on the TREC 8 collection

Using a corpus of queries for which the system's retrieval performance is known one can then apply automatic classification techniques in order to obtain predictions on the classification criterion (e.g. the difficulty of the query in terms of average precision obtained) for new queries not contained in the training corpus.

We have used several generic classification algorithms on different vector representations of the available information. The WEKA toolkit [Witten and Frank, 1999] provides a great number of different classifiers (such as decision trees, SVM, NaiveBayes, neural net, ...), but we have also experimented with other implementations, including CAL5 [Müller and Wysotzki, 1997], Dipol [Schulmeister and Wysotzki, 1997], or SVM*light* [Joachims, 1998].

Tree based algorithms present the advantage of constructing human readable and interpretable classification rules. This can provide interesting information about the classification process and provide useful insights for the refinement of the approach. However, given the limited amount of training data available, other classifiers (and particular support vector machines) in some cases obtain a better classification performance.

We have also done some tests with a classifier specialized in text categorization, the *Semantic Classification Trees (SCT)* [Kuhn and De Mori, 1995], which from a text collection (in our case the queries) constructs a tree based on (automatically extracted) regular expressions that serve to divide the corpus on the different branches of the tree.

The most interesting results have been obtained using SVM and decision tree algorithms.

Since we have a relatively large number of features to start with, we have investigated the effect of feature selection methods on classification performance. For decision trees, feature selection is implicit in the construction of the tree (the number of features used being limited by the number of non-leaf nodes). Additional reduction of the feature space prior to training the classifier has shown no positive effect. More about the features used by the decision trees is said in section 6.

For SVMs, a reduction of the feature space could be useful, if only for computational reasons. We have done some tests with different selection methods ranging from simple selection of the most strongly correlated features, selection of a number of features from each sub-group, to several methods such as PCA, CFS and others provided by the WEKA toolkit. Some of these would just select a subset of features, while others would e.g. create new features as linear combination of several of the original attributes.

While these selection methods may be beneficial in reducing the computational cost of classification, their effect on classification performance was inconclusive. In particular, possible overfitting when training the SVMs does not seem to depend strongly on the feature space (at least in our case). Results given in section 6 have been obtained without prior feature selection.

6. EXPERIMENTS

Our experiments have been conducted using the data from all systems whose results are available from the TREC 8 campaign, in which they performed the retrieval task for a set of 50 queries.

Based on the average precision of the retrieved documents, each query has been classified as "easy" or "hard". The split point between easy and hard queries had to be chosen quite arbitrarily (it would depend on the specific application setting); we have used the median value of the average precisions over the query collection for most of our experiments. This decision was made in order to obtain a balanced distribution of the training data into the two classes. Also, having an identical distribution for the different systems makes it possible to directly compare their classification performance.

The classification performances presented here were obtained using decision trees and SVMs implemented within the WEKA toolkit. The SVMs were used with a polynomial kernel and default settings. The decision trees are built using an implementation of the C4.5 algorithm. The default pruning settings yielded reasonably sized trees with depths between 3 and 7. Attempts at optimizing the parameters have been inconclusive.

As the number of queries is small, the *leave-one-out* method was initially used. The prediction accuracy for each system has to be compared to the accuracy obtainable solely on the basis of the class distribution, which with the chosen class separation is 50%. The features were evaluated for all participating systems of TREC 8 (as far as the available data allowed it). Detailed descriptions of these systems are available in the TREC 8 proceedings ([Voorhees and Harman, 1999]).

For the sake of brevity, the prediction accuracy is reported

in Table 2, for a representative set of systems that obtain a mean average precision (MAP) above a given threshold. For some systems, it appears that reliable predictions can be obtained, with e.g. a precision of 84% for the Okapi system on medium length queries (*ok8amxc*) using decision trees.

system	*MAP*	*DT*	*SVM*	*DT+SVM*	*coverage*
ok8amxc	32%	84%	56%	82%	66%
mds08a5	23%	82%	50%	79%	56%
mds08a2	24%	82%	56%	77%	70%
ok8asxc	28%	74%	68%	80%	70%
Sab8A4	26%	74%	70%	81%	72%
orcl99man	41%	74%	46%	66%	64%
INQ603	27%	72%	78%	80%	82%
Flab8at	29%	70%	52%	69%	58%
tno8d4	28%	70%	60%	73%	66%
att99atc	29%	68%	66%	70%	86%
uwmt8a2	27%	68%	72%	75%	80%
apl8p	32%	68%	30%	48%	42%
ric8tpx	27%	68%	44%	59%	68%
UT800	26%	68%	56%	71%	56%
weaver1	22%	68%	58%	71%	62%
Mer8Adtnd3	23%	66%	70%	85%	52%
iit99au2	20%	66%	72%	76%	74%
Mer8Adtd1	22%	66%	72%	77%	70%
Mer8Adtd2	22%	66%	72%	77%	70%
tno8d3	29%	66%	52%	67%	54%
ok8alx	32%	64%	66%	78%	54%
INQ602	25%	64%	72%	74%	76%
apl8c221	31%	64%	74%	76%	74%
ic99dafb	25%	62%	70%	70%	80%
acsys8aln2	26%	62%	70%	77%	60%
INQ604	28%	62%	70%	77%	60%
ric8dnx	24%	60%	66%	69%	70%
iit99ma1	41%	60%	70%	73%	66%
Sab8A3	25%	60%	72%	72%	72%
att99atde	32%	60%	74%	76%	66%
GE8ATD3	26%	58%	66%	66%	76%
apl8ctd	29%	54%	68%	65%	74%
iit99au1	23%	54%	70%	68%	68%
ric8dpn	26%	54%	70%	67%	72%
nttd8alx	28%	54%	70%	68%	68%
ibmg99b	26%	54%	72%	70%	66%
ric8dpx	27%	50%	66%	61%	72%
att99ate	28%	50%	74%	67%	72%
nttd8ale	29%	48%	70%	62%	74%
acsys8alo2	26%	46%	74%	69%	52%

Table 2: average prediction accuracy for a selection of TREC 8 participants

An example of a decision tree is given in Figure 2.

Here, each node corresponds to a test on a feature. Each branch is labeled with a test result. Each leaf is labeled with a classification result, along with the number of examples in the training set corresponding to the leaf's class label, followed by the number of examples pertaining to a different class.

The tree shown is the one calculated with the complete TREC 8 query collection as a training set. Using the same feature set, algorithm and settings, one obtains 84% of correct classification in a *leave-one-out* evaluation.

Using classification, we can take advantage of features that individually show only limited prediction capacity. Our approach is relatively robust, and we can obtain a relatively good classification even when removing some of the stronger features.

For example, when removing all measures related to the ratio of the retrieval system's document scores (*IR score (ratio N)*) which are prominently used in the example above,

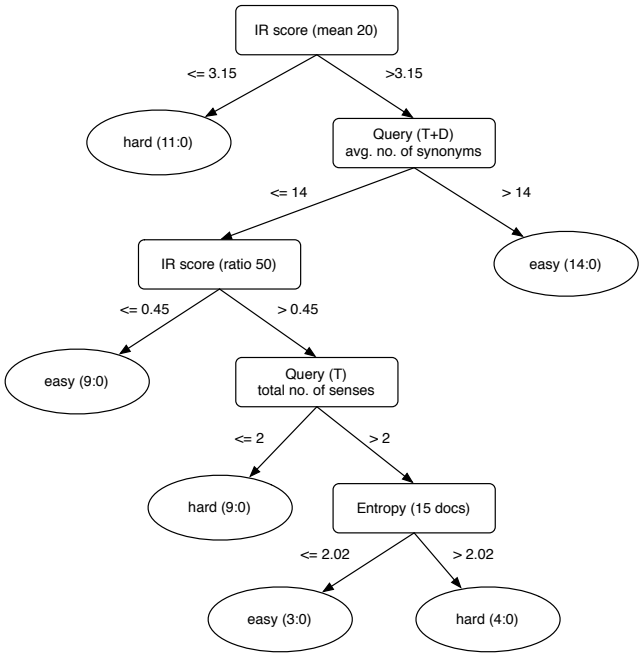

Figure 2: Decision tree related to ok8amxc

one obtains a new tree (Figure 3). Using leave-one-out, one obtains a classification accuracy of 76%. While this is a degradation from the performances using the full feature set, it is still a relatively good result, and shows how the combination of several weak features can sometimes compensate the lack of individually very strong predictors.

Table 3 lists all the features used, grouped into four sets characterizing:

- the features computed from the K top-ranked documents after retrieval;
- the features that depend on the IR ranking scores of the systems used in TREC 8 for which the results are available;
- the features that depend on the short query;
- the features that depend on medium length query.

The number of times each feature is used in the inferred classifiers (decision trees) for the different TREC 8 participants is also reported. It appears that there are frequently used features in each set, which suggests that each type of features is relevant to the classification task, despite the relatively weak correlation in some cases. Their combinations and statistical dependencies have been established by the learning process.

One can notice a great difference between the performances obtained using SVMs as compared to decision trees. In some cases, SVMs perform much better than DTs (e.g. for att99ate a classification precision of 74% is obtained with SVMs, whereas decision trees yield no better than random predictions), while the contrary is true for other systems (e.g. for ok8amxc, the precision is 84% with DTs compared to 56% using SVMs). We have therefore included in table 2 the classification precision obtained for those queries where both classifiers agree. The last column shows the percentage

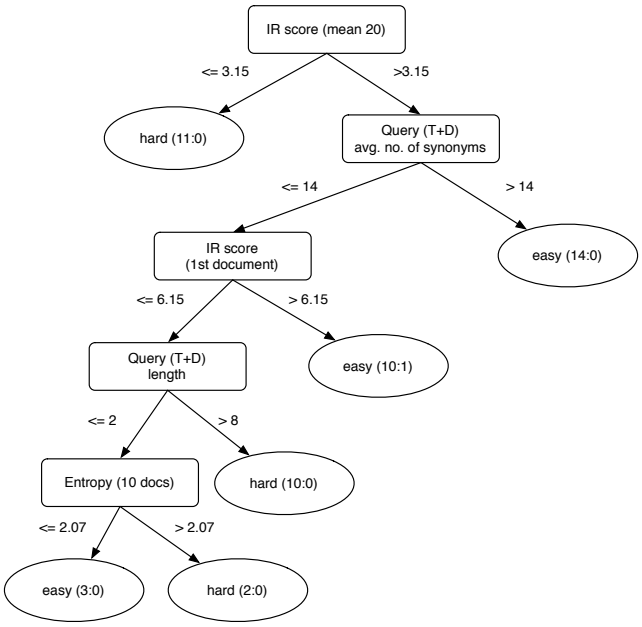

Figure 3: Decision tree related to ok8amxc, reduced feature set

	score	used by
documents	entropy (10 docs)	18
	entropy (15 docs)	20
	entropy (20 docs)	21
	entropy (5 docs)	28
	entropy (50 docs)	29
	mcs (10 docs)	14
	mcs (15 docs)	14
	mcs (20 docs)	18
	mcs (5 docs)	24
	mcs (50 docs)	22
IR ranking score	mean (10 docs)	13
	mean (15 docs)	2
	mean (20 docs)	7
	mean (5 docs)	16
	mean (50 docs)	6
	ratio (10th doc)	6
	ratio (15th doc)	8
	ratio (20th doc)	9
	ratio (5th doc)	16
	ratio (50th doc)	14
	score (1st doc)	33
	score (10th doc)	5
	score (15th doc)	5
	score(20th doc)	8
	score (5th doc)	25
	score (50th doc)	10

	score	used by
query title	abbreviations	
	technical dictionary	
	hyponyms (avg.)	17
	hyponyms (total)	10
	length (stopped)	
	length (raw)	45
	negations	
	senses (avg.)	15
	senses (de Loupy/Bellot)	47
	senses (total)	12
	synonyms (avg.)	8
	synonyms (total)	10
query title + description	abbreviations	
	technical dictionary	
	hyponyms (avg.)	18
	hyponyms (total)	13
	length (stopped)	27
	length (raw)	32
	negations	
	senses (avg.)	11
	senses (de Loupy/Bellot)	33
	senses (total)	26
	synonyms (avg.)	18
	synonyms (total)	5

Table 3: features with the number of systems for which they are used by the decision tree classifiers (MCS stands for *Mean Cosine Similarity*).

of queries covered by this consensus indicated as DT+SVM, i.e. the number of times the two classifiers agree.

In order to assess the generalization capacity of our approach and the resulting classifiers, an attempt has been made to train the classifiers with TREC 8 results and to test them with TREC 7 results. On the test, 66% classification accuracy is observed when using a classifier trained on INQ603 with TREC 8 data in order to predict the INQ503 system with TREC 7 data. The same procedure using Okapi yields 60% precision, whereas a classifier trained on mds08a5 (and strongly dependent on the ranking score) is not applicable for the mds98td system. It is important to keep in mind that these results are obtained not only on a different corpus but using a different retrieval system than used for training.

We have not yet investigated how closely related the systems are to their predecessors, which may contribute to the result variability. Classifiers using system dependent features cannot be expected to work well on data obtained with a very different version of a retrieval system. In spite of this, a prediction accuracy significantly greater than 50% has been observed. This is a promising result, suggesting that our approach is not too dependent on a specific corpus and retrieval setting.

7. DISCUSSION AND PERSPECTIVES

A method has been proposed for making predictions about the quality of retrieved documents, given a query and a repository. Experimental evidence has been provided that, for certain systems, predictions are quite reliable. Furthermore, experiments have shown that the reliability of performance prediction is not correlated with the performance value itself. For each system, classifiers have been automatically trained for classifying a query as "easy" or "hard". A set of specific features for this task was automatically selected by the training procedure. With the availability of more data, it would be interesting to investigate the possibility of adapting the features selected for a system to other, perhaps new, systems by including scores derived from the ranking functions used by systems exhibiting highly predictable performance.

In spite of the fact that participants to successive TREC campaigns vary in time as well as the systems used by each participant, for some systems good prediction performance is observed when classifiers are trained on the data of a campaign and tested on the data of another campaign. This suggests that prediction capability can improve as a greater amount of data becomes available for training the classifiers. Some tests were conducted with classifiers performing predictions of continuous values in order to directly estimate the average precision of a set of retrieved documents. Unfortunately, the results were not satisfactory, probably because of insufficient data available for estimating some parameters of the statistical model such as regression functions.

The classification performances reported in this article were obtained based on a collection of individually relatively weak features (for the most part). We have shown that relatively good classification can be obtained even when weakening the feature set by removing some of the most discriminant features. However, our approach obviously does depend on the features used and benefits from having highly predictive features available. As such, we expect a very positive effect from incorporating the different prediction

measures that have been proposed recently, in particular by [Cronen-Townsend et al., 2002] and [He and Ounis, 2004].

Another interesting use of classifiers is for deciding when to execute other processes, such as query expansion, after the execution of the basic retrieval process. A preliminary investigation has been conducted on learning when to perform *Blind Relevance Feedback* (BRF).

We have shown that for a large percentage of queries the application of BRF actually reduces the retrieval performance, despite a globally positive effect in terms of average precision over a complete set of queries. So far, we have obtained significantly above random precision on predicting positive or negative impact of BRF for a query, but the effect of selective BRF application on average precision has been inconclusive.

These results have been obtained with a feature set essentially designed for query difficulty prediction, it could be expected that incorporating features explicitly related to the query expansion process would significantly increase the quality of the predictions. On the other hand, we have only evaluated the results in terms of mean average precision over the query collection, while other measures might more appropriately reflect a possible gain for the particularly problematic queries.

It appears that automatically trained classifiers are also promising components for providing useful information for sequential decisions in an information retrieval system.

Practical applications can benefit from the proposed method. Among them, it is worth mentioning spoken document retrieval (e.g. [Jourlin et al., 1999]), for which a manual assessment of the relevance of each document can be particularly time consuming. A good estimation of the expected performance can thus result in a significant efficiency improvement.

8. REFERENCES

Amati, G., Carpineto, C., and Romano, G. (2004). Query difficulty, robustness and selective application of query expansion. In *Proceedings of ECIR'04*, Sunderland, UK.

Amitay, E., Carmel, D., Darlow, A., Herscovici, M., Lempel, R., Soffer, A., Kraft, R., and Zien, J. (2003). Juru at TREC 2003 - topic distillation using query-sensitive tuning and cohesiveness filtering. In *TREC-12 Proceedings*.

Banks, D., Over, P., and Zhang, N.-F. (1998). Blind men and elephants: Six approaches to trec data. *Information Retrieval*.

Callan, J. P., Croft, W. B., and Harding, S. M. (1992). The INQUERY retrieval system. In *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, pages 78–83.

Carpineto, C., de Mori, R., Romano, G., and Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19:1–27.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of SIGIR 2002*, pages 299–306. ACM Press.

de Loupy, C. and Bellot, P. (2000). Evaluation of document retrieval systems and query difficulty. In *Proceedings of "Using Evaluation within HLT Programs: Results and Trends"*, pages 31–38, Athènes, Grèce. http://www.limsi.fr/TLP/CLASS/ClassD43.pdf.

He, B. and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *Proceedings of SPIRE 2004*.

Joachims, T. (1998). Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

Jourlin, P., Johnson, S. E., Jones, K. S., and Woodland, P. C. (1999). Improving retrieval on imperfect speech transcriptions (poster abstract). In *Proceedings of SIGIR*, pages 283–284.

Kuhn, R. and De Mori, R. (1995). *The Application of Semantic Classification Trees to Natural Language Understanding*, volume 17, chapter 5, pages 449–460. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Müller, W. and Wysotzki, F. (1997). The decision-tree algorithm cal5 based on a statistical approach to its splitting algorithm. In *Machine Learning and Statistics: The Interface*, pages 45–65. R. Nakeiazadeh and C.C. Taylor.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1996). Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*.

Rorvig, M. (1999). Retrieval performance and visual dispersion of query sets. In *TREC-8 Proceedings*. http://trec.nist.gov/pubs/trec8/t8_proceedings.html.

Salton, G. (1989). *Automatic Text Processing – The Transformation, Analysis, and Retrieval of Information by Computer*. Addison–Wesley.

Schulmeister, B. and Wysotzki, F. (1997). Dipol-a hybrid piecewise linear classifier. In *Machine Learning and Statistics: The Interface*, pages 133–151. R. Nakeiazadeh and C.C. Taylor.

Spärck-Jones, K. (1999). Summary performance comparisons TREC-2 through TREC-8. In *TREC-8 Proceedings, Appendix B*.

Sullivan, T. (2001). Locating question difficulty through explorations in question space. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries*, pages 251–252. ACM Press.

Voorhees, E. M. (2003). Overview of the trec 2003 robust retrieval track. In *TREC-12 Proceedings*.

Voorhees, E. M. and Harman, D. (1996). Overview of the fifth text retrieval conference. In *TREC-5 Proceedings*. http://trec.nist.gov/pubs/trec5/t5_proceedings.html.

Voorhees, E. M. and Harman, D. (1997). Overview of the sixth text retrieval conference. In *TREC-6 Proceedings*. http://trec.nist.gov/pubs/trec6/t6_proceedings.html.

Voorhees, E. M. and Harman, D. (1999). Overview of the eighth text retrieval conference. In *TREC-8 Proceedings*.

Witten, I. H. and Frank, E. (1999). *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.