# Comparison Of Feature Construction Methods for Video Relevance Prediction

Pablo Bermejo[1], Hideo Joho[2], Joemon M. Jose[2], and Robert Villa[2]

[1] Computing Systems Dept., Universidad de Castilla-La Mancha, Albacete,Spain.
pbermejo@dsi.uclm.es
[2] Department of Computing Science, University of Glasgow, UK.
{hideo,jj,villar}@dcs.gla.ac.uk

**Abstract.** Low level features of multimedia content often have limited power to discriminate a document's relevance to a query. This motivated researchers to investigate other types of features. In this paper, we investigated four groups of features: low-level object features, behavioural features, vocabulary features, and window-based vocabulary features, to predict the relevance of shots in video retrieval. Search logs from two user studies formed the basis of our evaluation. The experimental results show that the window-based vocabulary features performed best. The behavioural features also showed a promising result, which is useful when the vocabulary features are not available. We also discuss the performance of classifiers.

**Keywords:** video retrieval, relevance prediction, feature construction.

## 1 Introduction

Multimedia databases have become a reality and as such, the need has arisen for effective multimedia information retrieval systems that work as accurately and fast as possible. Much research has been carried out on this problem from different points of views: ranking algorithms, feature construction, collaborative retrieval, etc., but unfortunately the performance of MIR systems is still far from that of text Information Retrieval (IR) due to the semantic gap: there is a discontinuity between low level visual features and the semantics of the query. Multimedia information retrieval systems usually use algorithms to create a ranking [14, 19] of results relevant to the text or image query submitted by the user. Some of these rankings have the problem of supporting just a very small number of features, such as those based on Term Frequency (TF) and Inverse Document Frequency (IDF) values. In the literature we can find several studies to select ([6]) or construct ([1]) features in order to improve the performance of ranking algorithms. However, there is limited work on the comparison of different groups of features to predict the relevance in Multimedia Information Retrieval (MIR), which is the main contribution of this paper. To do this, we project the retrieval problem into a classification problem and we work with databases constructed from logs created during two users studies. We consider

the information retrieval task as a supervised classification problem with a binary class attribute ("Relevant" and "Non Relevant"), where the documents which will be classified are a set of instances, each one representing a video shot described by a set of features. Formally the problem can be defined as a set of instances $C_{train} = \{(s_1, l_1), \ldots, (s_{|S|}, l_n)\}$, such that $s_i \in S$ is the instance which corresponds to the $i$th shot of the set of shots $S$, $l_j$ corresponds to the value of the class attribute that contains it and $L = \{Relevant, NonRelevant\}$ is the set of possible values for the class attribute. The goal is to build a classifier $c : S \rightarrow L$ to solve the prediction of a shots relevance; that is, the value of the class attribute for each instance. We will construct four kinds of features to find out which performs best and feature selection will be used to find which features (for each type of feature) are the most important.
We expect that our conclusions, found when performing classification, can also be used for the ranking process in a retrieval system, where classifiers can be mixed with ranking algorithms to improve the performance of retrieval systems.
The structure of this paper is as follows: Section 2 summarizes some work found in the literature related to feature construction and feature selection applied to information retrieval and classification. Section 3 presents our approach to creating different kinds of features, our selection method and how we perform the evaluation. The results for our experiments are shown in Section 4 and finally we present our main conclusions.

## 2 Related Work

The quality of the used set of features is of great importance for the classifier to achieve good performance [2]. This performance will depend on the individual relevance of each feature with respect to the class, relationship among features and the existence of features which influence negatively on the classifier.
It is possible to improve the quality of the available features by performing: (1) Feature Subset Selection, a widely studied task [10, 7] in data mining, which consists of reducing the set of available features by selecting the most relevant ones using filter metrics (statistical, distances, etc.) or wrappers (goodness of the classifier); and (2) Feature Construction: sometimes it is also possible to obtain new features with a higher quality from the original ones by computing some relation or statistic [9].

### 2.1 Feature Subset Selection

Feature Subset Selection (FSS) is the process of identifying the input variables which are relevant to a particular learning (or data mining) problem.
Though FSS is of interest in both supervised and unsupervised data mining, in this work we focus on supervised learning, and in particular in the classification task. Classification oriented FSS carries out the task of removing most irrelevant and redundant features from the data with respect to the class. This process helps to improve the performance of the learnt models by alleviating the effect

of the curse of dimensionality, increasing the generalization power, speeding up the learning and inference process and improving model interpretability.
In supervised learning FSS algorithms can be (roughly) classified in three categories: (1) embedded methods; (2) filter methods; and, (3) wrapper methods. By embedded methods we refer to those algorithms, e.g. C4.5, that implicitly use the subset of variables they need. Filter techniques are those that evaluate the goodness of an attribute or set of attributes by using only intrinsic properties of the data. Filter techniques have the advantage of being fast and general. On the other hand wrapper algorithms are those that use a classifier in order to asses the quality of a given attribute subset. Wrapper algorithms have the advantage of achieving a greater accuracy than filters but with the disadvantage of being more time consuming and obtaining an attribute subset that is biased toward the used classifier, although in the literature we can find some attempts to alleviate this problem [4].
In [6] some limitations of feature selection methods are stated when applied to ranking, which they consider as an optimization problem.

### 2.2 Feature construction

Several techniques [9] have been developed for the construction of new attributes through the application of operations over the available attributes. Attribute construction is most important when working with real world databases, which have not been created with thought to their application to data mining, and thus it is possible they do not contain attributes meaningful enough for beneficial use [5]. In attribute construction the main goal is to get a new attribute which represents the regularities of our database in a simpler way and thus makes the classification task easier [12]. Related to MIR, [1] created attributes which could represent the user behavior while searching data and thus be able to predict relevance for the user. Shots can also be represented by the visual features (texture, color layout, etc.) extracted from their keyframes. A lot of effort is currently being made to cross the semantic gap between query semantics and low level visual features (such as work on textual features [8]). To extract visual features from shots, several tools can be used, the most currently used being the MPEG-7 Visual Standard for Content Description [16]. Text created from transcript speech is another common way of representing shots. As stated in [22], although it can be used to gain good performance, it cannot be applied to all videos in general due to the lack of speech in some videos, or the fact that the speech does not relate to the visual content of the video.

## 3 Methodology

In order to learn how different kinds of constructed features affect relevance prediction, we have used data logs from two users experiments (see Section 4) and, from these logs, we have constructed final datasets (with different kinds

of features) used to evaluate classifiers. In this section we explain how our final datasets are constructed, the different kinds of features used, and how the classifiers are evaluated.

### 3.1 Datasets creation from user logs

We denote 'user study' to refer to an experiment in which several users tested a video retrieval system searching under different topics and conditions. A log file was created from each of the users studies [21] and [20] (see Section 4.2). Each log file contains verbose data explaining the actions each user performed (on which shots[3] actions were performed, the kind of action performed, timestamp, user condition, topic of search,...). For each query search, the user interacts with a set of shots. So, for each tuple ⟨search,user,condition,topic,shot⟩ a new instance is created from a log for the final dataset. Each instance in the final dataset consists of: tuple features (⟨search, user, condition, topic, shot⟩), features constructed to predict the class feature and which represent the shot, and the class feature itself. Class features are either Relevant or Non Relevant, and refers to the relevance of the shot in the corresponding tuple. For the same log, different final datasets have been constructed because different kind of features have been tested (see Section 3.2) to predict relevance and additionally, different kinds of relevance have been tested (see Section 4.1).

### 3.2 Kind of features used to predict relevance

As mentioned above, an instance in the final dataset will follow the pattern

$$Tuple\ Features,\ Relevance\ Prediction\ Features,\ Class\ Feature.$$

We have used four different kinds of *Relevance Prediction Features*: User Behavior Features, Object Features, Vocabulary Features and Windowed Vocabulary Features. Thus, for the logs from each user study four final datasets $DS1$, $DS2$, $DS3$ and $DS4$ are derived, where the four datasets contain the same values for $Tuple\ Features$ and $Class\ Feature$ but each one contains one of the four kind of features constructed.

Our User Behavior Features where designed similar to [1]. These features give information about how the user interacts with a document. In our case, the information is related to the actions the user performed through shots suggested by the information retrieval system after he/she ran a query under a concrete topic and condition. Behavior features used in this work are shown in Table 1 and they can be split into three groups: *Click-Through features*, which represent information about clicks the user performed on shots; *Browsing features*, which show different metrics about time spent on shots and *Query-Text features*, which count words in the current text query and make comparisons with other text queries. Note that the values for these features are computed for each tuple (⟨search,user,condition,topic,shot⟩) from the users studies logs.

[3] In Multimedia IR systems, retrieved documents are not the whole videos but shots, where a shot is one of the splits a video can be divided into.

**Table 1.** Behavior Features used to predict shots relevance

| Feature name | Description |
|---|---|
| ClickFreq | Number of mouse clicks on shot |
| ClickProb | *ClickFreq* divided by total number of clicks |
| ClickDev | Deviation of *ClickProb* |
| TimeOnShot | Time the user has been performing any action on shot |
| CumulativeTimeOnShots | *TimeOnShot* added to time on previous shots |
| TimeOnAllShots | Sum of time on all shots |
| CumulativeTimeOnTopic | Time spent under current topic |
| MeanTimePerShotForThisQuery | Mean of all values for *TimeOnShot* |
| DevAvgTimePerShotForThisQuery | Deviation of *MeanTimePerShotForThisQuery* |
| DevAvgCumulativeTimeOnShots | Deviation of *CumulativeTimeOnShots* |
| DevAvgCumulativeTimeOnTopic | Deviation of *CumulativeTimeOnTopic* |
| QueryLength | Number of words in current text query |
| WordsSharedWithLastQuery | Number of equal words in current query and last query |

**Table 2.** Object Features used to predict shots relevance

| Feature name | Description |
|---|---|
| Color Layout | vector containing 10 values |
| Dominant Color | vector containing 15 values |
| Texture | vector containing 62 values |
| Edge Histogram | vector containing 80 values |
| Content Based Shape | vector containing 130 values |
| Length | Time length of shot |
| Words | #words in Automatic Speech Transcription from shot audio |
| DifferentWords | #Different words in ASR from shot audio |
| Entropy | Shannon entropy of ASR from shot audio |

Object Features are not extracted from logs. They represent both *Low-Level Features* and *Metadata* and they are shown in Table 2. Using these features, the *Relevance Prediction Features* describe the shot appearing in *Tuple Features*. *Metadata* keeps information about length of shots and also information related to the Automatic Speech Recognition (ASR) from shots audio. Text transcripts from a shots' audio is filtered through a stop-words list and a Porter stemming filter ([13]), and then used to extract some statistics about the text.

Vocabulary Features are a bag of words created from the ASR. In this case the text is not used to compute statistics about the text, but to create a vocabulary of words to perform the task of text classification. The transcripted text is also filtered through a stop-word list and a Porter stemming filter. Then, the resulting text is transformed into Weka format using a tool based on Lucene [4]. For this kind of feature the video relevance classification becomes a problem of text classification.

It is expected that video relevance classification based on ASR works relatively well due to the fact that text has more descriptive power than, for example, low level visual features. However, in the literature some complaints about using ASR can be found, as in [22] where the authors state that some speeches might not have anything in common with their respective shots.

Finally, Windowed Vocabulary Features refer to a common technique in video retrieval systems which use ASR to create the results list. This uses the same

[4] http://lucene.apache.org/who.html

procedure performed when using Vocabulary Features but in this case the text used to construct the bag of words does not come only from the ASR of the corresponding shot but also from the $n$ previous shots in time and the later $n$ shots. This is call *n-Windowed ASR* and in our case we use a 6-Windowed Vocabulary. It is expected that 6-Windowed Vocabulary features perform better than creating a bag of words from only the ASR of a single shot.
When we use ASR to create a bag of words and evaluate using a bayesian classifier, we do not use Naive Bayes but the Naive Bayes Multinomial, which is recommended for text classification ([11]).

### 3.3 Evaluation method

Evaluation is performed without using the $Tuple\ Features$ so that the evaluation is totally free of context differentiation. In the case of using Behavior Features, which are continuous values and user dependent, it is difficult to construct a dataset with repeated instances. But, when using Object or Vocabulary Features, the same shot can appear in different tuples so the $Relevance\ Prediction\ Features$ are repeated; then we would have several repeated instances in the dataset where the class feature is sometimes set as *Relevant* and other times as $NonRelevant$. This contradiction is solved by deleting all repeated instances and setting the class feature to the most frequent value.
Datasets are evaluated by performing ten times a 10 cross validation (10x10CV) using three different classifiers (two statistical and one vector space based classifier): Naive Bayes, SVM (polynomial kernel, since this was the best configuration found) and kNN (k=1). As happens in information retrieval systems, datasets are very skewed due to a larger number of non relevant documents than relevant. So training sets are balanced by randomly deleting as many non relevant documents as needed so that the classifier is trained with the same number of relevant and non relevant documents. Although not a sophisticated way to balance datasets, the subject of balancing training data is outside the scope of this paper. For each 10x10CV, two metrics have been computed:

- TPrate(R) True Positive Rate for relevant documents represents the *recall* of relevant documents. The higher this metric is, the more relevant documents an information retrieval system will return.

$$TPrate(R) = \frac{\#relevant\ documents\ classified\ as\ relevant}{\#relevant\ documents} \quad (1)$$

- TPrate(NR) Although TPrate(R) is high, precision for relevant document could be low so TPrate(NR) would be low as well, what would make the system return many non relevant documents classified as relevant. Then, the main goal is to get both TPrate(R) y TPrate(NR) as high as it is possible.

$$TPrate(NR) = \frac{\#non\ relevant\ documents\ classified\ as\ non\ relevant}{\#relevant\ documents} \quad (2)$$

We have not used *Accuracy* to evaluate the classifiers because, although it is a standard metric used to evaluate the predictive power of classifiers, the tests sets

are so unbalanced that computing Accuracy is roughly the same as computing $TP_{NR}$. Although training sets are balanced, test sets are not: if a classifier always marks documents as belonging to the majority class value, accuracy would be incredibly high but documents belonging to the minority class values would never be correctly predicted. For information retrieval systems, documents belonging to minority class value (relevant documents) are what we want to predict correctly and so accuracy on its own is not an appropriate metric.
Finally, we performed an incremental wrapper-based feature selection based on ([3]), using the best configuration found in that study for the selection process. The goal for this selection is to find out what constructed features are most important for each kind of *Relevance Prediction Feature*.
This selection consists of an incremental wrapper-based feature subset selection (IWSS). First, a filter ranking by Symmetrical Uncertainty with respect to the class is constructed to rank all the available features. Then, from the beginning of the ranking to the end, the inclusion of each feature in the final subset of selected features is evaluated to decide if it must be added or not. This is presented in [15], and we use the best configuration found in [3] for this algorithm.

## 4 Experiments

Experiments were carried out on datasets constructed from logs obtained in two user studies. For each constructed dataset, four new datasets are derived using either User Behavior, Object, Vocabulary or 6-Windowed Vocabulary Features (see Section 3.2) to predict each shots' relevance. For each dataset, a 10x10CV is performed (using three different classifiers). For each evaluation, $TPrate_R$ and $TPrate_{NR}$ metrics are computed to compare classifiers capacity to predict relevance using different kinds of constructed features.

### 4.1 Kinds of relevance

We have used two sources of information to decide if a shot is relevant for a topic or not: Official Relevance and User Relevance. This means that for each final dataset, its evaluation is performed twice, once for each kind of relevance: (1) Official Relevance: Shots used in the users experiments belong to the TRECVid 2006 collection [17], which provides a list of the relevant shots for each topic based on the standard information retrieval pooling method of relevance assessment; and (2) User Relevance: In the user experiments, users could explicitly mark shots as relevant to the topic. In a dataset, a shot can be considered relevant if the user marks it as such.
One of the user studies this work is based on did not use the official TRECVid 2006 topics, so Official Relevance for that study cannot be used. Table 3 summarizes all the different evaluations performed for datasets obtained from each of the users studies (Collaborative and StoryBoard studies, introduced in Section 4.2).
Relevance predictions have a different meaning depending on the kind of features

**Table 3.** User studies used under different combinations of kind of features and kind of relevance.

| | **Official Relevance** | **User Relevance** |
|---|---|---|
| **User Behavior Features** | Collaborative | Collaborative & StoryBoard |
| **Object Features** | Collaborative | Collaborative & StoryBoard |

and relevance used. Predicting User Relevance using User Behavior Features can be seen as predicting explicit user feedback because users marked videos (or not) after interacting with them. Predicting Official Relevance using User Behavior Features predicts the relevance of a shot decided by a third group by actions users performed on the shots influenced by their perceptions. If we use Official Relevance when feeding our classifier with Object or Vocabulary Features values, we are assuming that low level features (as Color Layout) are meaningful enough to cross the semantic gap[5] [18] to high level concepts. Similarly, when predicting User Relevance using Object Features, some influence between low level features and metadata in user perception is assumed. When using Vocabulary Features, relevance prediction is similar to when Object Features are used, but in this case there is not a semantic gap, the problem becoming a text classification task.

### 4.2 User studies and datasets created

To create the datasets which will be evaluated, we have used logs coming from two users studies: the Collaborative study [21] and the StoryBoard study [20]:

- *Collaborative study.* In this study, users where grouped into pairs and searched for shots relevant to four Trecvid2006 topics under four different conditions: user A could see what user B was doing, user B could see user A, both users could see each other and, lastly, both users performed a search independently.
- *StoryBoard study.* In this study, users had to use two different interfaces (a common interface as baseline and a storyboard-style interface), to search for shots relevant to two different non-TRECVid topics.

As it can be seen in Table 3, 4 datasets were created from the Collaborative user study and 2 datasets from the StoryBoard user study. For each of these datasets, evaluation was performed using each of the four kinds of feature introduced in section 3.2.

### 4.3 Results

In this section we show the results obtained when performing classification with three different classifiers. Evaluation is performed over two databases created from 2 users studies. For each database, evaluation is performed using one of the four kinds of features, with the Collaborative users study represented twice, one for each kind of relevance.

[5] Distance between low level features (which have no meaning) and high level concepts.

**Table 4.** Results for datasets constructed from Collaborative users study - Official Relevance.

| | **Behavior** | | **Object** | | **Vocabulary** | | **W6-Vocabulary** | |
|---|---|---|---|---|---|---|---|---|
| | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ |
| NBayes/NBM | 0.42 | 0.87 | 0.76 | 0.49 | 0.61 | 0.47 | 0.80 | 0.52 |
| SVM | 0.52 | 0.71 | 0.65 | 0.64 | 0.47 | 0.62 | 0.68 | 0.72 |
| kNN | 0.68 | 0.69 | 0.79 | 0.51 | 0.44 | 0.65 | 0.53 | 0.92 |
| **mean** | **0.54** | **0.75** | **0.73** | **0.55** | **0.51** | **0.58** | **0.67** | **0.72** |

**Evaluations** We show the results of the performed evaluations in Tables 4, 5 and 6. If our aim is to get a $TP_R$ as high as possible without worrying about $TP_{NR}$, on average both Object and W6-Vocabulary Features are the best choice. If we are seeking for high $TP_{NR}$, both Behavior Features and W6-Vocabulary Features perform best. Since we need a good balance in both TP rates we can conclude that using the text from ASR of current shot and nearby shots to create the vocabulary is the best option to perform shot categorization. However, there are many videos in multimedia databases which have no text at all, or their text is not related to the contents, so in those cases another kind of feature would be needed to be constructed. Additionally it should be noted that Vocabulary Features created from a single shot have less predictive power than any other kind of constructed features presented in this work. If we do not compare results on average but taking into consideration classifiers on their own, we find that NBM for Vocabulary Features (windowed and not windowed) perform well for $TP_R$, although they perform with a lot of noise, besides working much faster than kNN. SVM is known to usually be the best classifier when performing text document classification; however, in this case the regarded problem is not that of text categorization so it is not very surprising that SVM has performed the worst. If our databases do not contain speech in their videos, it can be seen that Behavior Features work better than Object Features to predict Non Relevant documents, when it is important in an information retrieval system to get rid of noisy results. This means that using Behavior Features would create a system with less noise, but one which would also return fewer relevant shots.
As mentioned in Section 4.1, prediction of relevance has a different meaning depending on the kind of feature and relevance used in evaluation. So, if our aim is to construct an information retrieval system to collect relevant documents, behavior features would only make sense if we construct a collaborative information retrieval system, where the interactions performed by previous users through retrieved documents are stored and used in future searches from other users. While object features could be used in a standard information retrieval system and would always retrieve the same documents for the same queries.

**Feature Selection** Feature Selection has been performed based on ([3]), using the best configuration found in that study for the selection process.Selection has only been run on Behavior and Object Features, since selection on Vocabulary features would only return a set of words with no generalization power. Besides the Object Features set have a cardinality much higher than the Behavior

**Table 5.** Results for datasets constructed from Collaborative users study - User Relevance.

| | **Behavior** | | **Object** | | **Vocabulary** | | **W6-Vocabulary** | |
|---|---|---|---|---|---|---|---|---|
| | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ |
| NBayes/NBM | 0.55 | 0.82 | 0.67 | 0.41 | 0.73 | 0.45 | 0.71 | 0.48 |
| SVM | 0.62 | 0.70 | 0.57 | 0.56 | 0.53 | 0.64 | 0.54 | 0.66 |
| kNN | 0.70 | 0.71 | 0.60 | 0.47 | 0.47 | 0.58 | 0.63 | 0.48 |
| **mean** | **0.63** | **0.74** | **0.61** | **0.48** | **0.58** | **0.56** | **0.63** | **0.54** |

**Table 6.** Results for datasets constructed from StoryBoard users study - User Relevance.

| | **Behavior** | | **Object** | | **Vocabulary** | | **W6-Vocabulary** | |
|---|---|---|---|---|---|---|---|---|
| | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ |
| NBayes/NBM | 0.42 | 0.87 | 0.71 | 0.39 | 0.67 | 0.48 | 0.75 | 0.73 |
| SVM | 0.52 | 0.71 | 0.57 | 0.55 | 0.56 | 0.57 | 0.75 | 0.72 |
| kNN | 0.68 | 0.69 | 0.64 | 0.48 | 0.36 | 0.71 | 0.74 | 0.72 |
| **mean** | **0.54** | **0.75** | **0.64** | **0.47** | **0.53** | **0.59** | **0.75** | **0.73** |

Features set, so the final number of selected features in both should not be an issue.

In Table 7 we show the results of evaluations using only the selected features for each dataset. We can observe that feature selection decreases a bit the true positive rate for relevant shots, but on the other hand it significatively increases true positive rate for non relevant shots. This means that an information retrieval system would generate less noisy results than the same system not using feature selection. Since the number of relevant shots is so small, a tiny change in $TP_R$ is not significant, meanwhile these large increases in $TP_{NR}$ would mean a great change in the global accuracy.

In Table 8 we show the constructed features chosen by the incremental selection. With respect to Behavior Features, we can see that features constructed to represent statistics about clicks performed are the most frequently selected. This makes sense and can be expected, since clicks can be regarded as explicit feedback about the interests of the user.
With respect to Object Features, we indicate with a number the quantity of indexes selected from visual features vectors. We can see that visual features are more frequently selected than metadata computed for shots. Although this result has been a surprise it can be explained as being the effect of the larger number of visual features (if we count each index for each vector) compared to the four metadata features.

## 5 Conclusions and Future Work

We have tested four different kinds of features in a classification domain (with three different classifiers) where the class attribute is binomial with values {Relevant, Non Relevant}. All features have been tested on their own, without mixing different types. In order to not overfit, we have tested these features with databases constructed using logs from two different users studies and two different kinds

**Table 7.** Results after performing Incremental Wrapper-Based Selection

| | **Collaborative - Official R.** | | | | **Collaborative - User R.** | | | | **StoryBoard - User R.** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Behavior** | | **Object** | | **Behavior** | | **Object** | | **Behavior** | | **Object** | |
| | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ | $TP_R$ | $TP_{NR}$ |
| NBayes | 0.48 | 0.88 | 0.67 | 0.59 | 0.62 | 0.83 | 0.33 | 0.76 | 0.15 | 0.96 | 0.59 | 0.52 |
| SVM | 0.24 | 0.92 | 0.62 | 0.64 | 0.58 | 0.74 | 0.62 | 0.55 | 0.40 | 0.81 | 0.47 | 0.56 |
| kNN | 0.72 | 0.64 | 0.65 | 0.57 | 0.76 | 0.70 | 0.55 | 0.54 | 0.78 | 0.78 | 0.59 | 0.48 |
| **mean** | **0.48** | **0.81** | **0.65** | **0.60** | **0.65** | **0.76** | **0.50** | **0.62** | **0.44** | **0.85** | **0.55** | **0.52** |

**Table 8.** Selected features when performing Incremental Wrapper-Based Selection

| **Behavior** | Official R. | User R. | User R. | **Object** | Official R. | User R. | User R. |
|---|---|---|---|---|---|---|---|
| ClickFreq | x | x | x | Color Layout[10] | | | |
| ClickProb | x | x | | Dominant Color[15] | 1 | | 5 |
| ClickDev | x | x | x | Texture[62] | | 1 | |
| TimeOnShot | | | x | Edge Histogram[80] | 10 | 2 | 5 |
| CumulativeTimeOnShots | | | x | Content Based Shape[130] | 1 | 1 | |
| TimeOnAllShots | x | | | Length | | x | |
| CumulativeTimeOnTopic | | | | Words | | | |
| MeanTimePerShotForThisQuery | x | x | x | DifferentWords | | | |
| DevAvgTimePerShotForThisQuery | | | | Entropy | | x | |
| DevAvgCumulativeTimeOnShots | | | x | | | | |
| DevAvgCumulativeTimeOnTopic | | x | x | | | | |
| QueryLength | x | | x | | | | |
| WordsSharedWithLastQuery | | | x | | | | |

of relevance.
Our main conclusion is that Windowed Vocabulary Features perform, on average, better than the rest, where a good performance is regarded as the best possible balance between $TP_R$ and $TP_{NR}$.
Feature selection helps to decrease noise while insignificantly losing a little performance for relevant documents. Additionally, the most relevant features have been identified for User Behavior Features and Object Features.
For future work, it would be interesting to study the effect on relevance prediction when mixing different kind of constructed features, and also testing how different contexts affect the classifier's performance. It would also be interesting to apply our techniques to re-ranking the output of an information retrieval system, to investigate potential improvements in performance.

## 6 Acknowledgments

This work has been partially supported by the JCCM under project (PCI08-0048-8577), MEC under project (TIN2007-67418-C03-01), FEDER funds, MIAUCE project (FP6-033715), and SALERO Project (FP6-027122).

## References

1. E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM Press.
2. R. Bekkerman, A. McCallum, and G. Huang. Automatic categorization of email into folders: Bechmark experiments on enron and sri corpora. Technical report, Department of Computer Science. University of Massachusetts, Amherst., 2005.

3. P. Bermejo, J. Gámez, and J. Puerta. On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. In *IPMU'08: Proceedings of the 12th Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2008.
4. M. J. Flores, J. Gámez, and J. L. Mateo. Mining the esrom: A study of breeding value classification in manchego sheep by means of attribute selection and construction. *Computers and Electronics in Agriculture*, 60(2):167–177, 2007.
5. A. A. Freitas. Understanding the crucial role of attributeinteraction in data mining. *Artif. Intell. Rev.*, 16:177–199, 2001.
6. X. Geng, T.-Y. Liu, T. Qin, and H. Li. Feature selection for ranking. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 407–414, New York, NY, USA, 2007. ACM.
7. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
8. P. Howarth and S. M. Rüger. Evaluation of texture features for content-based image retrieval. In *CIVR*, pages 326–334, 2004.
9. Y.-J. Hu. *Constructive induction: covering attribute spectrum In Feature Extraction, Construction and Selection: a data mining perspective*. Kluwer, 1998.
10. H. Liu and H. Motoda. *Feature Extraction Construction and Selection: a data mining perspective*. Kluwer Academic Publishers, 1998.
11. A. McCallum and N. K. A comparison of event models for naive bayes text classification. In *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.
12. F. Otero, M. Silva, A. Freitas, and J. NIevola. Genetic programming for attribute construction in data mining. In *Genetic Programming: Proc. 6th European Conference (EuroGP-2003).*, 2003.
13. M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.
14. S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994), Gaithersburg, USA*, 1994.
15. R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recogn.*, 39:2383–2392, 2006.
16. T. Sikora. The mpeg-7 visual standard for content description-an overview. 11(6):696–702, June 2001.
17. A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.
18. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.
19. K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.
20. R. Villa, N. Gildea, and J. M. Jose. Facetbrowser: a user interface for complex search tasks. In *ACM Multimedia 2008 (In press)*, 2008.
21. R. Villa, N. Gildea, and J. M. Jose. Joint conference on digital libraries. In *A Study of Awareness in Multimedia Search*, pages 221–230, June 2008.
22. R. Yan and A. G. Hauptmann. Co-retrieval: A boosted reranking approach for video retrieval. In *CIVR*, pages 60–69, 2004.