Multimed Tools Appl (2012) 60:203–231
DOI 10.1007/s11042-011-0818-x

Simulating the future of concept-based video retrieval under improved detector performance

Robin Aly · Djoerd Hiemstra · Franciska de Jong · Peter M. G. Apers

Published online: 31 May 2011

Abstract In this paper we address the following important questions for concept-based video retrieval: (1) What is the impact of detector performance on the performance of concept-based retrieval engines, and (2) will these engines be applicable to real-life search tasks if detector performance improves in the future? We use Monte Carlo simulations to answer these questions. To generate the simulation input, we propose to use a probabilistic model of two Gaussians for the confidence scores that concept detectors emit. Modifying the model's parameters affects the detector performance and the search performance. We study the relation between these two performances on two video collections. For detectors with similar discriminative power and a concept vocabulary of around 100 concepts, the simulation reveals that in order to achieve a search performance of 0.20 mean average precision (MAP)—which is considered sufficient performance for real-life applications—one needs detectors with at least 0.60 MAP. We also find that, given our simulation model and low detector performance, MAP is not always a good evaluation measure for concept detectors since it is not strongly correlated with the search performance.

Keywords Concept-based retrieval · Simulation · Performance prediction · Concept detection

1 Introduction

Content-based video retrieval currently mainly focuses on the improvement of concept detectors [26]. On the other hand there is research on developing retrieval models to combine the output of concept detectors to fulfill information needs of

R. Aly (✉) · D. Hiemstra · F. de Jong · P. M. G. Apers
University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands
e-mail: r.aly@ewi.utwente.nl

users. Although concept-based retrieval is generally a promising retrieval paradigm, the search performance of currently available engines is often too low for large-scale application in real-life. Clearly, the overall search performance heavily depends on detector performance. Therefore, it is desirable to answer the following research questions: (1) *What is the impact of detector performance on the performance of concept-based retrieval engines, and* (2) *will these engines be applicable to real-life search tasks if detector performance improves in the future?* This paper investigates the use of Monte Carlo Simulations to answer this question.

Hauptmann et al. [12] were the first to use a simulation-based approach to predict the achievable performance of concept-based video retrieval engines. In this work, noise is introduced into the known occurrences and absences of concepts by randomly flipping their states. Therefore, detectors are assumed to be binary classifiers which only differentiate between concept occurrence and absence. We argue that this approach can be improved upon since most retrieval engines today employ confidence scores or a probability measure based on these scores as document representations. The reason is that errors in binary classifications are frequent and the information of "shot x contains concept y with a confidence of z" needs to be exploited. For example, the concept *US-Flag* is often useful for answering the query "President Obama". However, the corresponding detector might classify no shots as containing the concept *US-Flag* but may find few shots more likely to contain a *US-Flag* than others, which could be exploited. Therefore, the simulation approach in this paper generates confidence scores for each shot and concept in a concept vocabulary which can then be transformed into probability measures as well as classifications.

This paper follows the Monte Carlo Simulation approach by Metropolis and Ulam [16] to predict the search performance of retrieval engines when the detector performance increases. The simulation approach requires a function which calculates a quantity of our interest based on a set of inputs. Here, this function will be the mean average precision (MAP) achieved by a retrieval engine in a search task and the inputs are the detector confidence scores in the collections. The application of the Monte Carlo Simulation approach allows us to split the stated research question into two sub-questions:

1. *How to model concept detectors?* In order to answer this question, we assume that confidence scores of detectors are independent from each other. Furthermore, we make the assumption that the confidence scores are normally distributed in the set of shots where the concept occurs and likewise where they are absent (the positive and negative class). This assumption is supported by studies of actual detector outputs in this paper and by Hastie and Tibshirani [11] for the output of general classifiers. Therefore, our probabilistic model consists of a set of concept detectors where each detector is defined by the parameters of two Gaussian distributions.
2. *What search performance to expect from a retrieval engine for a given detector model?* In order to answer this question, we use the probabilistic model and a collection with known concept occurrences to generate a set of randomized confidence scores. On this output, we then execute a retrieval run using a given retrieval engine and subsequently calculate the search performance in MAP. This process is repeated multiple times to calculate the expected MAP of the retrieval engine given the probabilistic model.

With the answer to these two questions, we can gradually change the parameters of the detector models to improve the detector performance and investigate the effect on the expected search MAP. From the development of the expected search performance compared to the detector performance we can predict the applicability of a retrieval engine in the future, the answer to the investigated research question.

The remainder of this paper is structured as follows: In Section 2 we give an overview of related work. Section 3 describes the probabilistic model which is used to simulate the detectors. In Section 4, we describe the simulation setup. In Section 5, we investigate the results of the simulation on a collection with concept annotations and relevance judgments. Section 6 concludes this paper with a summary and a discussion.

2 Related work

In this section we present an overview of related work to the object domains: prediction of multimedia search performance, concept detection and Monte Carlo Simulations.

2.1 Multimedia search performance prediction

Simulations to analyze the effects of the performance of a content analysis process on search performance have been used in various sub-fields of content-based multimedia retrieval. Croft et al. [9] use simulations to determine the effects of word-error-rates in optical character recognition systems on search performance. Witbrock and Hauptmann [31] simulate a varying word-error-rate of an automatic speech recognition system, to investigate its influence on the search performance of a spoken document retrieval engine.

Hauptmann et al. [12] were the first to use a simulation-based approach to investigate the feasibility of concept-based search performance. In their work, a detector is assumed to be a binary classifier. As a retrieval function they use a linear combination of concept occurrences: $score_q(d) = \sum_i w_i \ f_i(d)$. Here, $score_q(d)$ is the retrieval score of shot d, w_i is a concept specific weight and $f_i(d) \in \{-1, 1\}$ is the label of concept i in shot d. The weights w_i are independently set for each query. The weight setting which optimizes the average precision is found by solving a bounded constrained global optimization problem [33]. The search performance with realistically set weights is assumed to achieve 50% of the performance with optimal settings. For the simulation, concept labels of shots are randomly flipped until the precision-recall break even point is reached. We argue that this approach can be improved because current retrieval engines use confidence scores and a uniform break even precision-recall point assumes the same performance among all detectors, which is unrealistic.

Similar to the approach in this paper, Toharia et al. [30] simulate confidence scores to study the usefulness of concept-based retrieval. A concept from an annotated collection is assumed to have a score of -1 if it is absent and 1 if it occurs. For the simulation, noise is introduced by adding or subtracting to a certain percentage of P shots a value A, which lessens the confidence of the detector about the occurrence of a concept. As a retrieval function a weighted sum of the confidence scores is assumed where the weights are determined by users. The simulation is carried out by varying

the percentage P from 0 to 0.5 and A from -0.5 to 0.5. While this approach also simulates the influence of confidence scores on the search performance, it does not take into account that the confidence scores for shots where a concept is absent could be higher then some confidence scores for shots where the concept occurs. Therefore the mean average precision of the produced detector output is always 1.00. Our simulation can be considered an improvement as the confidence scores are set more realistically.

There are also other aspects than the detector performance which influence the search performance which are not covered in this paper: Christel and Hauptmann [8] investigate the general helpfulness of single concepts to retrieval. Furthermore, the effects of concept vocabulary size on the search performance by randomly including or excluding a growing number of concepts has been widely studied [12, 25].

2.2 Concept detection

The majority of current concept detectors are using Support Vector Machines (SVM) [27, 35]. Therefore, we adapt our model to the characteristics of SVMs: a SVM is trained on vectors of low-level features from positive and negative examples. The training phase selects so-called support vectors which specify a hyper-plane separating instances of the positive and negative class. During the prediction phase, the confidence score of the new shot, represented by its low-level feature vector $\mathbf{x}$, is calculated as follows, see also [18]:

$$o(\mathbf{x}) = \sum_i y_i \alpha_i k(\mathbf{x}, \mathbf{x_i}) + b$$

Here, $\mathbf{x}$ is the feature vector of the new shot, y_i the label and α_i the weight for the ith support vector $\mathbf{x_i}$ and $k(\cdot, \cdot)$ a kernel function between two feature vectors. b is constant. The result of the function $o(\cdot)$ is the confidence score of the SVM for the new shot. For simplicity, we drop the notation as a function and write o instead of $o(\cdot)$ in the following. If the SVM is treated as a binary classifier, a decision criterion is used to derive a classification from o. However, as mentioned above, in video retrieval it is more common to use the confidence score which can be demonstrated by their use in the influential works by [32] and [26]. Furthermore, confidence scores are used in the evaluation of concept detectors in the TRECVid workshop [24]. The reason is that classification errors are commonly too high, especially for rare concepts. The confidence score can be seen as an indicator of the likelihood that the current shot contains the concept in question.

For many applications it is useful to use a normalized probability for the class membership of a shot instead of an uncalibrated confidence score. Hastie and Tibshirani [11] propose that the confidence scores are normally distributed in the positive and negative class. Together with a prior these parameters can be used to calculate the posterior probability of encountering a concept after observing a confidence score o. However, the resulting posterior probability function $P(C|o)$ is not monotonically increasing with o. This is unrealistic, since some negative instances with lower confidence scores than others would have a higher posterior probability. Platt [18] proposes a method which instead fits a sigmoid function to the confidence scores of training examples which can then be used as a posterior probability function. The sigmoid function has the advantage of being always monotonic. In this

paper we will use a modified version of Platt's fitting algorithm suggested by Lin et al. [15], which is also used in many SVM implementations, for example libSVM [7].

2.3 Monte Carlo simulation

This paper proposes a simulation approach based on Monte Carlo Simulation [16]. The term Monte Carlo Simulation is used for a variety of different methods. In this paper, we use it for a general procedure to calculate the expected value of a function given the probabilistic model of its inputs. A Monte Carlo Simulation can be described as a procedure consisting of the following steps:

1. The definition of a probabilistic model of the inputs to the simulation.
2. Random generation of a concrete set of inputs using the model.
3. Execution of the function using the generated inputs.
4. Repetition of 2. and 3. to produce multiple results.
5. Average the results of the individual computations into the final result.

The results of this simulation is guaranteed to converge with an increasing number of repetitions to the expected function value (performance measure), based on the probabilistic model.

In this paper, the objective function is the search performance of a retrieval model in terms of MAP which depends on the confidence scores as inputs. The probabilistic model defines the distribution of the confidence scores. The random generation of inputs therefore randomly draws confidence scores from their distributions and the search performance of the retrieval model is calculated.

3 Detector model and simulation process

In this section we describe the probabilistic model proposed in this paper and the simulation process.

3.1 Detector model

In this section we describe the probabilistic model of confidence scores, which later is used for the randomization of confidence scores. Figure 1 shows the confidence score histograms of the two concepts *Anchorman* and *Outdoor* for the positive and negative class from a base line detector set, described by Snoek et al. [27]. The different score ranges and the resulting probability density magnitudes are caused by the detector's ability to discriminate between positive and negative examples. We propose that the densities for the positive and negative class of both concepts have roughly a Gaussian shape. This shape was also proposed by Hastie and Tibshirani [11] for the distribution of decision scores for general classifiers. We conducted a χ^2 goodness-of-fit test, see [29] for a definition, to assess the fit of these distributions. The test revealed that 31 of the 101 detectors in the vocabulary can be accepted as being a Gaussian distribution at a significance level of 0.05. Out of the 31 concepts which were accepted, 22 had more than 800 training examples, which suggests that the Gaussian shape would also become evident for other concepts if we had more training examples. Furthermore, Sangswang and Nwankpa [22] argue that

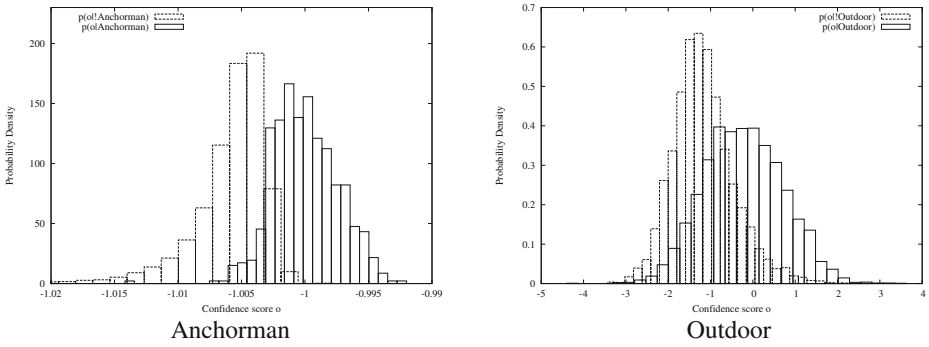


Fig. 1 Confidence score distributions of two concepts of the MediaMill detector set, see [27]

a non-perfect fitting shape of a model only increases the variance of the Monte Carlo Simulation, but still allows a trustworthy estimation of the expected search performance.

Given these observations, we define a probabilistic model of a detector set: we assume that the confidence scores of different detectors for a single shot are independent from each other and that they are normally distributed in the positive and negative class. Each concept C has a different prior probability $P(C)$. To keep the probabilistic model simple, we assume that all concepts share the same mean μ_1 and standard deviation σ_1 for the positive class plus the mean μ_0 and the standard deviation σ_0 for the negative class. Note, that this assumption is strong and certainly does not hold in reality, see for example Fig. 1. However, because we focus here on the principle influence of the detectors on the search performance we leave the exploration of a more realistic model, which investigates different parameter settings for each detector, to future work. Also, while the investigation of different means and deviations is important, we argue that the intersection of the areas under the probability density curves has a much higher influence on the performance than the absolute ranges of the confidence scores. The smaller the area of the intersection the better the detector is. Our model can adequately simulate this effect by either moving the means apart or by varying the standard deviation of the positive and negative class.

Figure 2 shows the model of a single detector. We plot the posterior probability of observing the concept given the confidence score using two different priors, one of $P(C) = 0.01$ and one for $P(C) = 0.60$. Considering a confidence score of $o = 15$ the posterior probability for a concept with the prior of 0.60 is close to certainty ($P(C|o) \simeq 1$) while for a concept with a prior of 0.01 it is undecided (50%)—with all other parameters equal. Therefore, our model does not have the limitation that all detectors have the same performance as assumed by Hauptmann et al. [12].

3.2 Posterior probability measure

As noted by Platt [18], the assumption of two Gaussians for the negative and positive class can lead to unwanted effects for the posterior probability function, namely that the function can be non-monotonic. Figure 3 shows the posterior probability

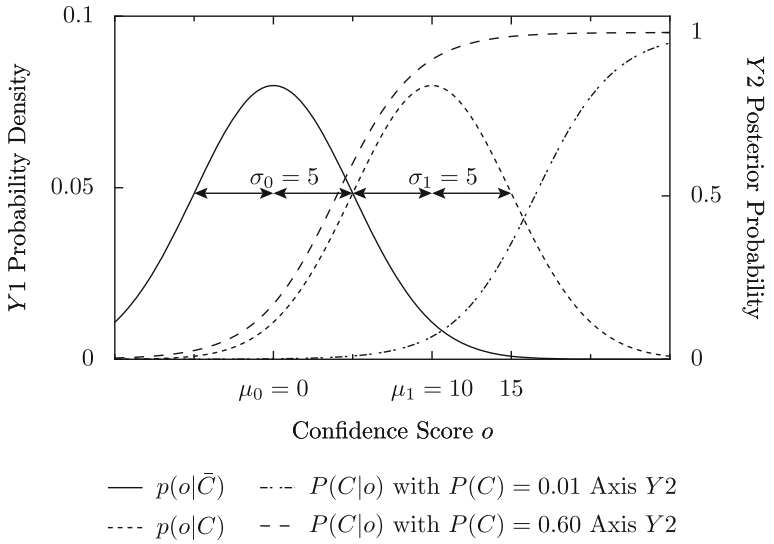


Fig. 2 Probabilistic detector model consisting of two Gaussians for the positive and negative class together with two possible posterior probability functions for different priors

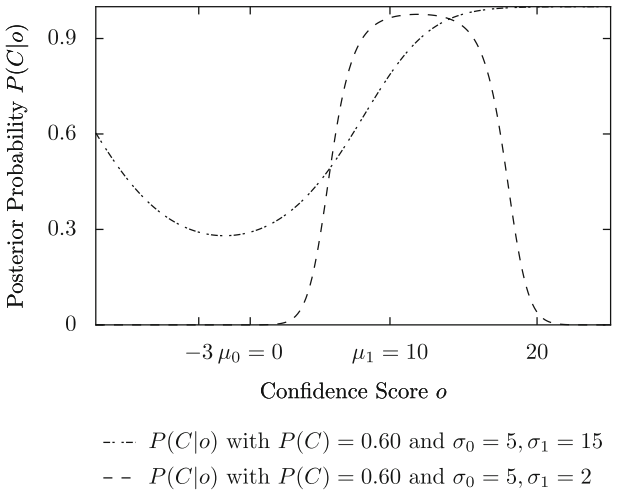


Fig. 3 Non-monotonic posterior probability functions, resulting from using two Gaussians

functions of two hypothetical concept detectors defined by the standard formula for posterior probabilities:

$$P(C|o) = \frac{p(o|C)P(C)}{p(o|C)P(C) + p(o|\bar{C})P(\bar{C})}$$

We see that with a standard deviation of $\sigma_1 = 15$, the posterior probability decreases for increasing confidence scores with $o < -3$. Furthermore, the posterior probability

function with $\sigma_1 = 2$ assigns to shots with confidence scores higher than 20 a posterior probability of practically 0. This contradicts our intuition and the definition of SVM based detector (where the positive and negative classes should be linearly separable). To prevent this effect we use an improved version of the algorithm from Platt [18], suggested by Lin et al. [15], to fit the parameters of a sigmoid function which is used as a model posterior probability function to the confidence scores of a set of training examples. The sigmoid function is defined as follows:

$$P(C|o) = \frac{1}{1 + \exp(A\,o + B)} \tag{1}$$

Here, A and B are the two parameters of the sigmoid function. Note, because the algorithm from Lin et al. [15] depends on the number of training examples, retrieval models which depend on the probabilistic output of (1) could suffer from a poorly fitted posterior function. To investigate the influence of the quality of the fit on the search performance, we use S hypothetic training examples for the fitting process

Algorithm 1 Algorithm for a simulation run. NR: number of repetitions, S: sample size for sigmoid fitting. $\mu_0, \sigma_0, \mu_1, \sigma_1$: model parameters

Data: Annotated Collection $\mathcal{D}$, Vocabulary $\mathcal{V}_C$
Input: $NR, S, \mu_0, \sigma_0, \mu_1, \sigma_1$
Result: Randomized collection

// Randomize Prior Estimate
foreach *Concept* C *in Vocabulary* $\mathcal{V}_C$ **do**
  Calculate $P(C)$ from annotations in $\mathcal{D}$
  generate $\lceil S\,P(C) \rceil$ positive training examples from $N(\mu_1, \sigma_1)$
  generate $S - \lceil S\,P(C) \rceil$ negative training examples from $N(\mu_0, \sigma_0)$
  determine A_C and B_C according to Lin et al. [15], given the training examples
end
// Randomize Detection Output
for *Repetition* $i \in [1..NR]$ **do**
  foreach *Shot* s *in Collection* $\mathcal{D}$ **do**
    foreach *Concept* C *in Vocabulary* $\mathcal{V}_C$ **do**
      if C *occurs in* s **then**
        draw o from $N(\mu_1, \sigma_1)$
      else
        draw o from $N(\mu_0, \sigma_0)$
      end
      // Calculate Posterior according to Platt [18]
      $P(C|o) = \frac{1}{1+exp(A_C\,o+B_C)}$
      // Transform to Binary Value
      if $P(C|o) > 0.5$ **then**
        $C = 1$
      else
        $C = 0$
      end
    end
  end
  Calculate Detector Performance $DMAP_i$
  Search Run with Retrieval Model
  Calculate Search Performance $SMAP_i$
end
Report Detector and Search MAP
$DMAP = \frac{\sum_i DMAP_i}{NR}$ $\quad SMAP = \frac{\sum_i SMAP_i}{NR}$

and randomly generate $\lceil S\ P(C) \rceil$ confidence scores from the positive class and $S - \lceil S\ P(C) \rceil$ from the negative class of concept C. The results of this investigation can be found in the Experiment in Section 5.5.

3.3 Simulation process

In this section we describe the actual simulation process which is described in pseudo-code in Algorithm 1. The algorithm uses an annotated collection (which carries 0/1 labels for each concept in each shot). The input parameters of the algorithm are the means μ_0, μ_1 and standard deviations σ_0, σ_1 of the positive and negative class and the number of training examples S to fit the posterior function. A Gaussian distribution with mean μ and standard deviation σ is denoted as $N(\mu, \sigma)$.

From the annotated collection we calculate the prior probability $P(C)$ of a concept occurring in the collection. We then generate confidence scores for the positive and negative class using the prior probability and a total of S training examples. Now, we use the algorithm described by Lin et al. [15] to fit the sigmoid posterior probability function to the generated training examples. After the determination of the sigmoid parameters we iterate over all shots in the annotated collection. For each shot we determine for each concept C in the vocabulary $\mathcal{V}_C$ whether it occurs and draw a random confidence score o from the corresponding normal distribution. Afterwards, we calculate the posterior probability of this concept in the shot using the sigmoid function with the previously determined parameters A_C and B_C. For retrieval models which use binary classifications we assume a positive occurrence if the posterior probability is above 0.5. This is justified by decision theory, see for example [6].

After the randomization, we determine the detector MAP of the detector output ($DMAP_i$). We then execute a search run for each retrieval model using the randomized collection. We then evaluate the resulting ranking using relevance judgments to obtain the search MAP ($SMAP_i$) for this run. This process is repeated NR times to rule out random effects and the expected detector performance (DMAP) and search performance (SMAP) are calculated.

4 Simulation setup

This section describes the setup of the simulation, describes their results and ends with a discussion.

4.1 Collections and concept vocabularies

We use the TRECVid 2005 (tv05d) and the TRECVid 2007 (tv07d) development collections plus the corresponding query sets [23] for our simulations. The relevance

Table 1 The collections used in the simulations

Identifier	Videos	Shots
tv05d	141–277	43,907
tv05dd	141–238	30,630
tv05dt	239–277	13,277
tv07d	001–110	18,120

Table 2 The concept vocabularies used in the simulations

Identifier	Collection	Concepts	Reference
mm101	tv05d	101	[27]
vireo374	tv05d	374	[14]
tv070809bw	tv07d	65	[5, 28]

judgments on the TRECVid 2005 development collection were kindly provided by Rong Yan formerly at Carnegie Mellon University [34]. The relevance judgments on the TRECVid 2007 development collection were provided by us [1]. To prevent over-fitting when performing realistic concept selections (see below) we divide the tv05d collection according to the MediaMill Challenge setting [27] into the sub-collections tv05dd (development) and tv05dt (test). Due to its limited size, we do not split the tv07d collection and hence do not run simulations with realistic weights on this collection. Table 1 shows statistics over the used collections. Table 2 summarizes the concept vocabularies which were used for the simulations. (1) the mm101c vocabulary [27] which comprises 101 concepts; (2) the vireo374 vocabulary [14] which comprises 374 concepts;[1] and (3) the tv070809bw vocabulary consists of the concepts annotated during the collaborative annotation efforts of the TRECVid participants during the years 2007–2009 which was coordinated by Ayache and Quénot [5] plus an additional black and white concept [28].

We use a Java-based (pseudo) random number generator[2] following a standard algorithm described by [19]. In order to make the simulations reproducible we use the open source software[3] described in [1] with a random seed of 994158012. For each parameter setting we generate $NR = 25$ sets of confidence scores. The simulations showed that the simulation results did not change anymore after this number of repetitions. We use in the following the common term MAP, instead of emphasizing every time that the number is actually obtained as an average over 25 runs.

To give an indication of the quality of the detectors we report the achieved detector MAP on the provided annotations. We used the same standard cut-off level of 2,000 as done for the High Level Feature task in TRECVid [23] to maintain comparable to other results. However, this cut-off level sometimes leads to counter intuitive results because some frequent concepts occur more than 2,000 times and consequently even a perfect detector would have an average precision of less than 1.0. Therefore, in such cases we assumed a maximum of 2,000 shots in which the concept occurred.

4.2 Investigated retrieval models

Concept-based retrieval models usually consist of two parts. First, a ranking function which uses a set of selected concepts to calculate a ranking score value for each document in a collection. Second, a method to select this set of concepts and their weights for a query (referred to as concept selection and weighting). In the following,

[1] We use the vireo374 vocabulary because it excluded seldom occurring concepts from the well-known LSCOM [17] vocabulary, of which it is a subset; we however do not use the published detector output.

[2] http://www.ee.ucl.ac.uk/~mflanaga/java/PsRandom.html

[3] http://detectsim.sourceforge.net/

we describe the considered retrieval functions for two considered retrieval tasks followed by the investigated concept selection and weighting methods.

The following retrieval functions for video shot retrieval are considered in this paper (Table 3 shows their mathematical definition). First, the pointwise mutual information weighting scheme (PMIWS) by Zheng et al. [36], which calculates a sum of the concept occurrence probabilities, weighted by the pointwise mutual information that a concept occurs. Second, the Borda–Count model which originates from election theory and considers the ranks of the confidence scores, see Donald and Smeaton [10] for the application to information retrieval. Third, the binary independence model (BIM) by Robertson et al. [20] which uses the parallel of a concept occurring in a shot and a book being indexed with an index term. Finally, our probabilistic ranking framework for unobservable binary events (PRFUBE), see [2], which calculates the expected probability of relevance score given the uncertain knowledge about the concept occurrence.

We now define the weights used for the above retrieval models. The PMIWS, BIM and PRFUBE model require the occurrence probability $P(C|R)$ of a concept C given relevance R as a weight, which is the number of relevant shots which contain the concept C divided by the number of relevant shots. The probability of a concept given irrelevance, used by the BIM model, is defined accordingly. The prior probability of concept occurrence $P(C)$ is the number of shots containing the concept C divided by the collection size. The prior probability of relevance $P(R)$ is the number of relevant shots divided by the collection size. The reader can think of $P(C|R)$ as steering the importance of a concept in a query and $P(C)$ or $P(C|\bar{R})$ as determining the general importance of a concept. For the Borda–Count model, we assume the Mutual Information between a concept C_i and relevance as an ideal weight w_i for this concept. The Mutual Information can be defined using the three parameters $P(C)$, $P(R)$ and $P(C|R)$, see [3].

In this paper, we focus on the investigation of the detector performance influence on retrieval functions. Therefore, we limit ourselves to alternatives to supply the $P(C|R)$ weight and select concepts. First, we perform one simulation using oracle weight settings, where we use the concept annotations and relevance judgments and determine the optimal weights by counting. Second, we perform another experiment of a realistic scenario where we use the Annotation-Driven Concept Selection method, proposed in [3], which is based on an annotated development collection. To use this concept selection method without introducing over fitting effects we use the

Table 3 Overview of the video shot retrieval models used in the simulations

Video shot retrieval models		
Identifier	Description	Definition
PMIWS	Pointwise mutual information weighting scheme, see [36]	$\sum_i^n \log\left(\frac{P(C_i\|R)}{P(C_i)}\right) P(C_i\|o_i)$
Borda–Count	Rank based, see [10]	$\sum_i^n w_i \, rank(P(C_i\|o_i))$
BIM	Binary independence model, see [20]	$\sum_i^n c_i \log\left(\frac{P(C\|R)(1-P(C\|\bar{R}))}{P(C\|\bar{R})(1-P(C\|R))}\right)$
PRFUBE	Probabilistic ranking framework for uncertain binary events, see [2]	$\prod_i^n \frac{P(C_i\|R)}{P(C_i)}) P(C_i\|o_i) + \frac{P(\bar{C}_i\|R)}{P(\bar{C}_i)}) P(\bar{C}_i\|o_i)$

(n: number of selected concepts, c_i: most probable concept occurrence)

collection tv05dt for weight estimation and later execute the search only on tv05dd. We also have to set the number of concepts which should be used for the search. As this is not the focus of this paper we tried multiple numbers of concepts with a maximum of 20 together with the results of using all concepts in the vocabulary.

4.3 Performed simulations

As our goal is to study the influence of the detector performance over the different model parameters we vary them piecewise to see the effect of each parameter on the overall search performance. In the following we describe each performed simulation and the characteristics of the set of detectors resulting from it:

- Model Coherence: here, we set the mean and variance individually according to the values from a real detector set. We then investigate, whether the simulated performances are comparable with the results of the real detectors.
- Changing the mean of the positive class: detectors with a higher difference between their means have better performance, which can for example originate from the use of more discriminative low-level features for which the detector is built.
- Changing the standard deviation of the positive class: detectors with a higher standard deviation have more extreme results in the positive class. For increasingly many shots in which the concept occurs is the detector nearly certain of the occurrence (has a high confidence score) while at the same time for many other shots in which the concept occurs the detector has low confidence scores.
- Changing the standard deviation of the negative class: detectors with a higher standard deviation of the negative class has more extreme results in shots where the concept does not occur. For many shots the detector is increasing certain that the concept (rightfully) does not occur. At the same time, for many other shots he has an increasing confidence that the concept does occur.
- Changing the number of training examples: we increase the number of training examples for fitting the sigmoid posterior probability function, which investigates the influence of the fit quality, caused by a small number of training examples, on the search performance.
- Predictive power of detector MAP: the change of the standard deviation of the detectors causes the search performance to over proportionally decrease compared to the detector MAP. Therefore, we investigate the predictive power of the detector MAP for the search MAP in a separate simulation.

5 Simulation results

In this section, we describe the results of the simulations carried out in this paper.

5.1 Model coherence

In this section, we investigate the coherence of the proposed probabilistic model with the MediaMill Challenge detector set by Snoek et al. [27]. In this experiment, we first fit the model parameters to the confidence scores of the detector set. We expect that the average detector performance is close to the performance of the real detectors.

Table 4 Simulation results for investigating the model coherence (collection tv05dt)

Measure	Expected result	Simulation max	Real detectors
Detector MAP	0.13	0.16	0.15
Search MAP	0.06	0.11	0.10

However, the search performance of the simulation is not necessarily equal to the real search performance, because of the random distribution of confidence scores in relevant shots. On the other hand, the real search performance should also not be too far off from the search performance produced by the model.

First, we train detectors for the mm101 vocabulary using the features provided by the Challenge Experiment 1 [27] using the tv05dd collection and then perform the evaluation on the tv05dt collection. Because we are only interested in the influence of the detector performance on the search performance we only use PRFUBE with oracle weights for 10 concepts per query. We estimate the model parameters from the confidence scores of the real detectors and set the mean and standard deviation of the positive and negative class individually. We calculate the mean and the deviation for the class $x \in \{0, 1\}$ and concept c by maximum likelihood estimation [21]:

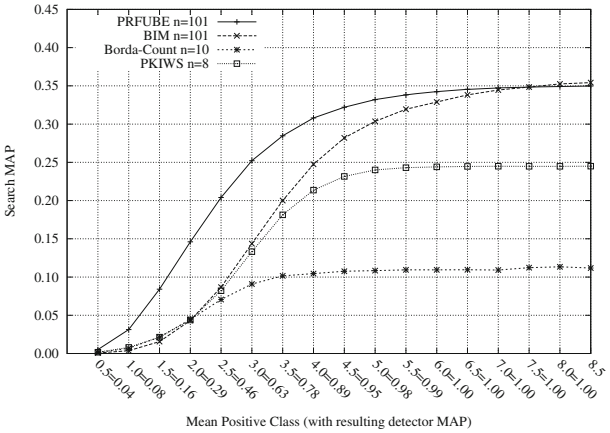
$$\mu_{xc} = \frac{\sum_{i=1}^{N_{xc}} o_{ic}}{N_{xc}}, \quad var_{xc} = \frac{\sum_{i=1}^{N_{xc}} (o_{ic} - \mu_{xc})^2}{N_{xc} - 1}, \quad \sigma_{xc} = \sqrt{var_{xc}}$$

Here, N_{xc} is the number of samples of the class x and o_{ic} is the observed confidence score of shot i and concept c. We perform 30 simulation runs. The results of the coherence study are shown in Table 4. We see that the average simulated detector performance of 0.13 MAP is lower than the one of the real detectors with 0.15 MAP. However, the maximal performance achieved by the simulation—among the 30 repetitions—exceeds the performance of the real detector, achieving 0.16 MAP. A possible explanation of the lower simulation performance is the correlation of confidence scores among many shots (≈2,000) in the tv05dt collection, which suggests that they are near duplicates or are highly similar. Because the proposed probabilistic detector model generates the confidence scores independently, the simulation is not able to capture these dependencies. However, we argue that the inclusion of the correlation of confidence scores in the probabilistic model is also not necessarily desirable because near duplicates or highly similar shots can be handled outside the search procedure. The search performance of our model is also lower compared to the real detectors, which means that the confidence scores of used concepts where higher in the real detector set. However, three simulation runs achieve an equal or higher search performance to the real detectors. We conclude that the proposed probabilistic detector model is sufficiently realistic to explain a current, realistic retrieval setting, except the handling of near duplicates and collections with many similar shots.

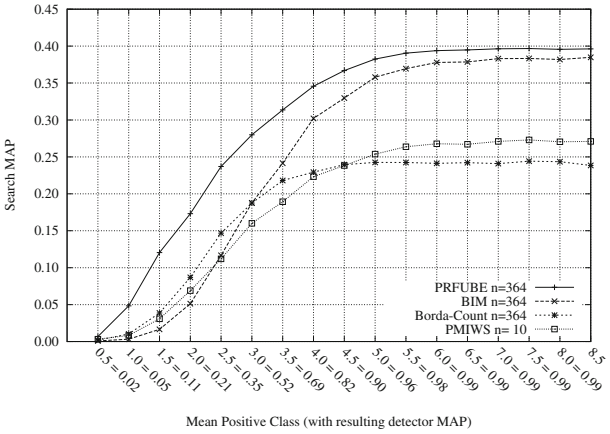
5.2 Changing the mean of the positive class

Oracle weights Figure 4a shows the results of the simulation which increases the mean of the positive class using the mm101 vocabulary. The y-axis shows in all following figures the achieved search MAP of the depicted retrieval models. The

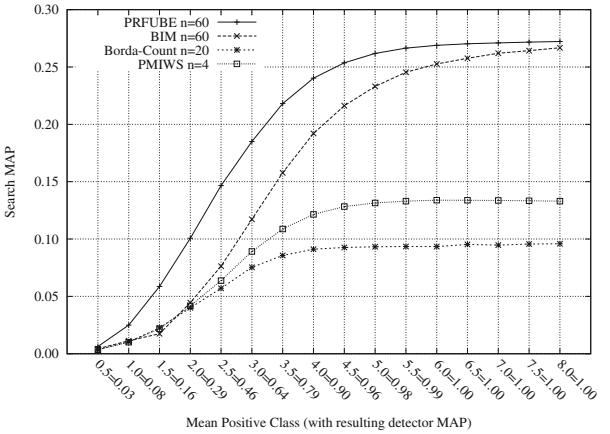
Fig. 4 Oracle weights: changing the mean of the positive class μ_1 ($\mu_0 = 0.0$, $\sigma_0 = 1.0, \sigma_1 = 1.0$)



(a) Tv05d collection, mm101 vocabulary



(b) Tv05d collection, vireo374 vocabulary



(c) Tv07d collection, tv070809bw vocabulary

x-axis shows the mean μ_1 together with the detector MAP which resulted from this setting, the remaining parameters are kept constant, see Fig. 4. An increasing μ_1 leads to an increase of the detector performance. The performance of all concept-based retrieval models increases with a growing detector performance. From a positive mean of $\mu_1 = 8.5$ onwards the detectors can be considered perfect classifiers. The PMIWS model reaches with ten concepts its best search performance of 0.15 MAP. Borda–Count also performs best when limited to the ten most influential concepts and achieves an optimal performance of 0.27 MAP. The BIM model has a slow start and only reaches a search performance of 0.05 MAP at $\mu_1 = 2$ which corresponds to a detector performance of 0.29 MAP. Afterwards, its performance increases faster than the two previously mentioned models and reaches at $\mu_1 = 8.5$ a performance of 0.36 MAP. PRFUBE consistently shows a better search performance than all other retrieval models and achieves at $\mu_1 = 8.5$ a search performance of 0.35 MAP. The BIM and PRFUBE retrieval models performed best with the usage of all concepts in the vocabulary.

Figure 4b shows the results of the simulation using the vireo374 vocabulary and oracle weights. The results are similar to the usage of the mm101 vocabulary. Notable is that this time the PMIWS model achieves a better search performance than Borda–Count. The reason is probably the existence of more only positive influential concepts—which can be exploited by the PMIWS model. The higher number of concepts allows PRFUBE to increase its search performance to 0.39 MAP.

Figure 4c shows the search performance when changing the mean of the positive class in the tv07d collection using the tv070809bw vocabulary and oracle weights. The results are similar to the ones of Fig. 4a and b. The PRFUBE model shows the best performance and reaches at a mean μ_1 of 3.00 a search performance of 0.185 MAP.

The expected search performance is the mean search performance of the possible search performances resulting from different parameter settings. Therefore, we also investigate how far the search performances are apart for a given parameter setting. Figure 5 shows the same performance graph as Fig. 4a with the minimum and maximum performance of the models in the $NR = 25$ simulation runs, see Algorithm 1. We used minima and maxima instead the standard deviation which is usually employed in similar graphs, because the standard deviations were too small

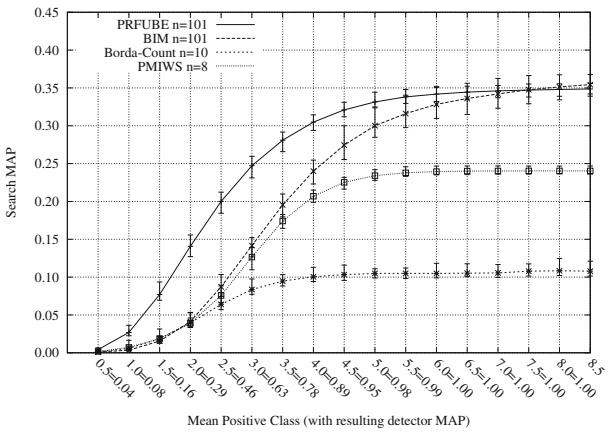
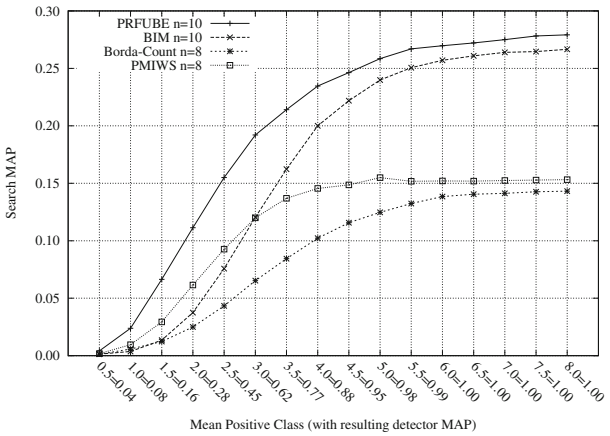


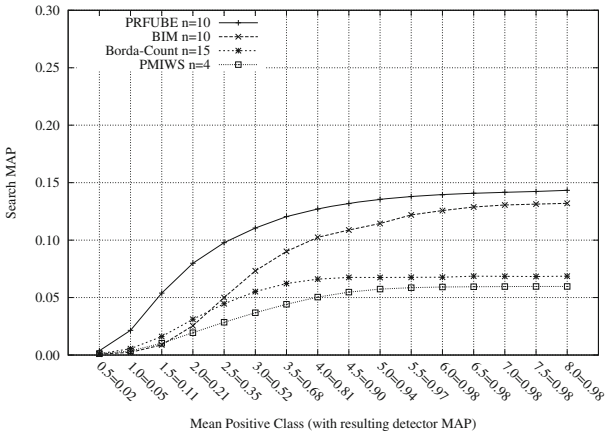
Fig. 5 Oracle weights: changing the mean of the positive class μ_1 with the minimum and maximum search performance of the simulation ($\mu_0 = 0.0$, $\sigma_0 = 1.0$, $\sigma_1 = 1.0$, $N = 25$). The standard deviation of the search performance is not visible to the human eye

to be displayed in the graph. Note however that minima and maxima only give a general impression of the distribution of search performances, which we belief is sufficient given the results. The PMIWS model and the Borda–Count model have similar search performance distributions. Especially for detector performances in the interval $[0.29 - 0.95]$ MAP, the ranges of search performances for the BIM model and the PRFUBE model are bigger. We repeated the study of performance minima and maxima for the other simulations of increasing the mean of the positive class presented in this paper which did not yield qualitative differences.

Realistic weights Figure 6a and b show the search performance on the tv05dd collection when the weights are realistically estimated from the tv05dt collection using the Annotation-Driven Concept Selection method, proposed in Aly et al. [3]. Figure 6a shows the simulation results of the search performance using the mm101 vocabulary. Because the weights are now estimated by a realistic concept selection



(a) Tv05dd collection, mm101 vocabulary



(b) Tv05dd collection, vireo374 vocabulary

Fig. 6 Realistic weights: changing the mean of the positive class μ_1 ($\mu_0 = 0.0$, $\sigma_0 = 1.0$, $\sigma_1 = 1.0$)

method, the search performance is lower for all retrieval models. The performance of the retrieval models relative to each other stays approximately the same.

Figure 6b shows the simulation results of the retrieval models using the vireo374 vocabulary. All models perform worse compared to the alternative of using the mm101 vocabulary. A likely explanation is that with a growing concept vocabulary the chance of selecting poor concepts—or setting wrong weights—increases.

5.3 Changing the standard deviation of the positive class

Figure 7a shows the results of changing the standard deviation of the positive class using oracle weight settings. We fix all other model parameters as follows: $\mu_0 = 0, \sigma_0 = 1, \mu_1 = 3$. An increase of the standard deviation of the positive class increases the uncertainty and therefore the difficulty of the search. Consequently, all retrieval models show a lower performance with an increasing standard deviation. The PMIWS model stabilizes at 0.05 MAP. The other three retrieval models Borda–Count, PRFUBE and BIM show a continuous performance loss. The retrieval model PRFBUE has the highest performance decrease but continues to show the best overall performance.

Figure 7b shows the increase of the standard deviation with weights from the realistic concept selection method. Here, PRFUBE stays around 0.03 MAP above all other retrieval models. The PMIWS model shows a worse performance than Borda–Count and BIM.

Figure 7c shows the result of changing the standard deviation of the positive class in the tv07d collection using oracle weights. The result is similar to the ones from the tv05d collection, see Fig. 7a and b. The PRFUBE performs best over the whole range of standard deviations, only this time the performance improvement compared to the next best model BIM is more distinct. The two confidence score dependent ranking functions PMIWS and Borda–Count show a similar performance over the changes of the standard deviation.

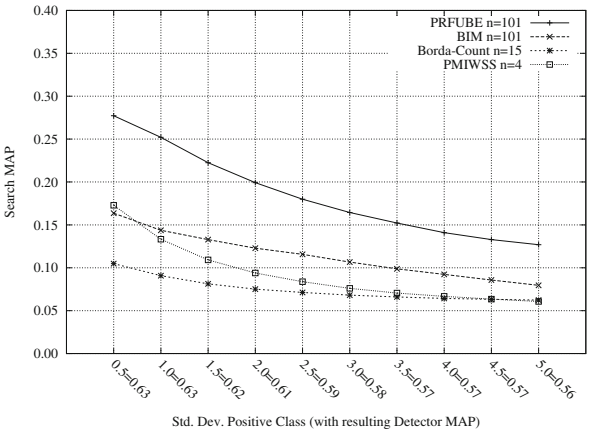
5.4 Changing the standard deviation of the negative class

Figure 8 shows the results of changing the standard deviation of the negative class using oracle weight settings. We fix all other model parameters as follows: $\mu_0 = 0, \mu_1 = 3, \sigma_1 = 1$. Similar to the change of the standard deviation of the positive class, an increase of the standard deviation of the negative class increases uncertainty. The detector performance quickly decreases with an increasing standard deviation of the negative class. For example, a modest change of the standard deviation from $\sigma_0 = 1$ to $\sigma_0 = 1.5$ results in an absolute detector performance decrease of 0.43 (65% relative). Similarly, the search performance of all retrieval models drops. All retrieval models except the PRFUBE model show a search performance of below 0.02 MAP for a standard deviation of $\sigma_0 > 2$. The search performance of the PRFUBE model decreases slower and reaches a search performance of 0.02 MAP at $\sigma_0 = 3.5$.

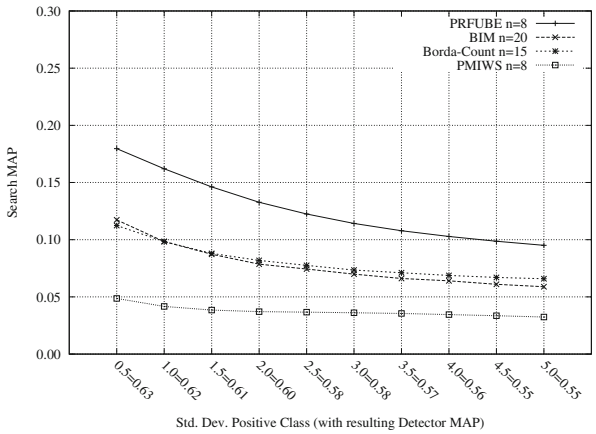
5.5 Sigmoid fitting

Figure 9 shows the results of an increasing number of training examples S used in the fitting procedure for the posterior probability function. Here, we used the

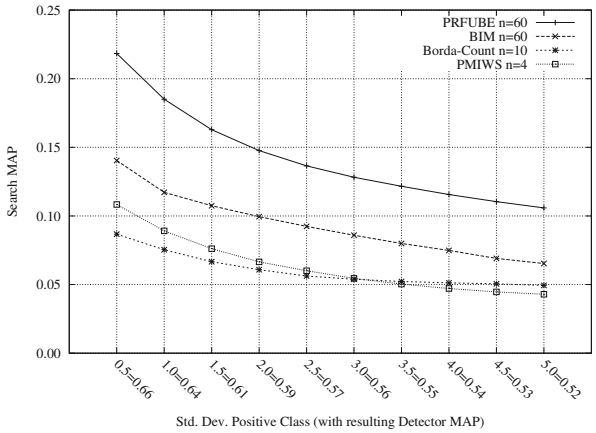
Fig. 7 Changing the standard deviation of the positive class σ_1 ($\mu_0 = 0.0, \sigma_0 = 1.0, \mu_1 = 3.0$)



(a) Tv05d collection, mm101 vocabulary, oracle weights



(b) Tv05dd collection, mm101 vocabulary, realistic weights



(c) Tv07d collection, tv070809bw vocabulary, oracle weights

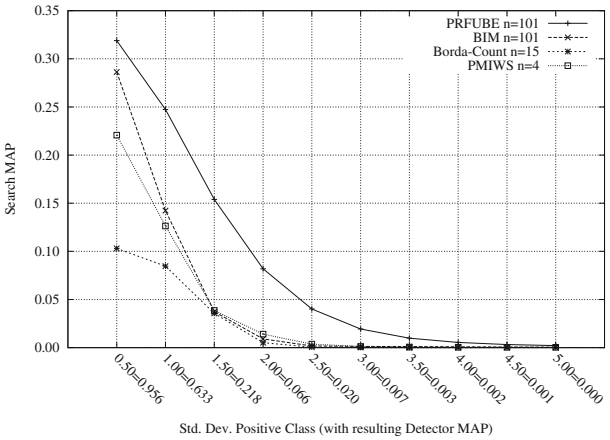


Fig. 8 Changing the standard deviation of the negative class (collection tv05d, mm101 vocabulary, σ_0 ($\mu_0 = 0.0$, $\mu_1 = 3.0, \sigma_1 = 1.0$)

mm101 vocabulary together with oracle concept weight settings. The x-axis shows the number of training examples S on a log-scale because smaller training sizes are of higher interest. Except of small random effects, the Borda–Count model shows constant performance because it does not depend on the probabilistic output.

For the BIM and PMIWS retrieval models the search performance decreases until a number of training example of $S = 100$. The reason is that for a small amount training examples of $S = 5$ the minimum number of one positive training example over represents the positive class. Therefore, the posterior probabilities are strongly biased towards higher values and the posterior probabilities and the positive classifications rise. Because the false negatives are the biggest problem for the BIM model its performance decreases. The same holds for the PMIWS model because the ranking formula only considers the probability of concept occurrence in relevant shots, see [2]. With an increasing number of $S > 100$ training examples this effect diminishes. The performance of the BIM and PMIWS models stabilizes

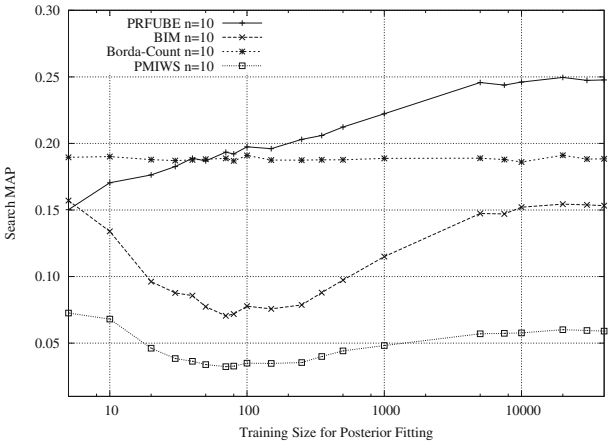


Fig. 9 Influence of training size S using oracle weights (collection tv05d, mm101 vocabulary, $\mu_0 = 0.0, \sigma_0 = 1.0$, $\mu_1 = 3.0, \sigma_1 = 1.0$)

after $S = 5{,}000$ because of increasingly accurate estimates of the parameters for the sigmoid function.

The PRFUBE improves its search performance linearly from 0.15 MAP using 5 training examples to 0.24 MAP with 5,000 examples. Beyond 5,000 samples it stays approximately constant. It is the positively affected by over-estimated posterior probabilities of a small training example size.

5.6 Predictive power of detector MAP

Figure 10 shows the scatter plot resulting from 200 random combinations of mean μ_1 and standard deviation σ_1 of the positive class, with $0.5 \leq \mu_1 \leq 8$ and $0.5 \leq \sigma_1 \leq 5$. We show the search performance of the PRFUBE model, since it performed best by the previous tests. The x-axis depicts the detector MAP and the y-axis depicts the search MAP. With a detector MAP of above 0.75 the detector MAP and the search MAP are strongly correlated. However, with a lower detector MAP the correlation decreases. For a detector performance between 0 and 0.40 MAP is the Pearson correlation of the search and detector performance only 0.50. The influence of this low correlation can be demonstrated with an example: at a detector performance of 0.16 MAP (which is the performance of best performing detectors at TRECVid), we measured a search performance of 0.07 MAP for detectors with $\mu_1 = 1.46$ and $\sigma_1 = 1.06$. However, we also measured a search performance of only 0.02 MAP for the same detector performance with $\mu = 0.74$ and $\sigma_1 = 1.76$. This indicates that detector MAP, as currently used to measure the performance of detectors in TRECVid is not a good evaluation measure. A good detector does not only need sufficient MAP, it also needs a low standard deviation for the scores of the positive class.

5.7 Discussion

In Fig. 4 we investigate the effects of concept detectors which get on average more confident about the correct occurrence of concepts. The PRFUBE combination model consistently performs best while the BIM model achieves approximately the

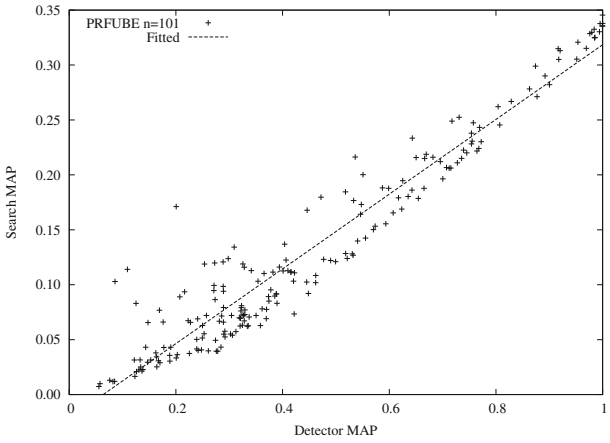


Fig. 10 Predictive power of detector MAP for search performance (200 randomly selected pairs of $\mu_1 \in [0.5 : 8]$ and $\sigma_1 \in [0.5 : 5]$, collection tv05d, vocabulary mm101, $\mu_0 = 0.0, \sigma_0 = 1.0$)

same performance after a slow start. The Borda–Count is often better than the BIM model in lower performance regions but stabilizes at a lower performance level when the detectors approach certainty. The PMIWS model can not gain as much performance from the increased detector performance. However, with a detector performance close to certainty it sometimes performs better than Borda–Count.

Using realistically estimated weights instead of oracle weights, see difference between Figs. 4 and 6, shows that the concept selection and weighting method has a strong influence on the search performance. The relative performance difference of the retrieval models compared to oracle weights is in general stable, only the PMIWS model performs now worse than the Borda–Count model.

In Fig. 7 we show that the increase of the standard deviation of the positive class decreases performance in all retrieval models. The effect of such an increase is that the distribution of confidence scores in the positive class is more extreme. In other words, the detectors emit for more shots either extremely high or low confidence score. As a result, the probability that a relevant shot in which a concept occurs is assigned a very low confidence score increases. This effect is also visualized in Fig. 11 which shows the expected detector precision and recall for the mm101 vocabulary with three different standard deviations. We see that the precision with a higher standard deviation is initially higher at lower recall levels. However, for higher recall levels this effect is reversed and an increasing number of shots with concept occurrences are not found among the first 2,000 shots.

The influence of an increasing standard deviation in the negative class causes higher detector performance and search performance decreases than for the change of the standard deviation of the positive class, see Fig. 8. With an increasing standard deviation of the negative class, a detector is increasingly confident about the absence of concepts in many shots. On the other hand, for other shots the detector has a high confidence about the occurrence of a concept where the concept actually does not occur. Figure 12 shows the effect of changing the standard deviation of the negative class on the precision recall curve. Similar to the increase of the standard deviation in the positive class, the precision decreases with an increasing standard deviation of the negative class for low recall levels. However, additionally the precision at low recall levels also decreases quickly. The reason is that there are usually many more

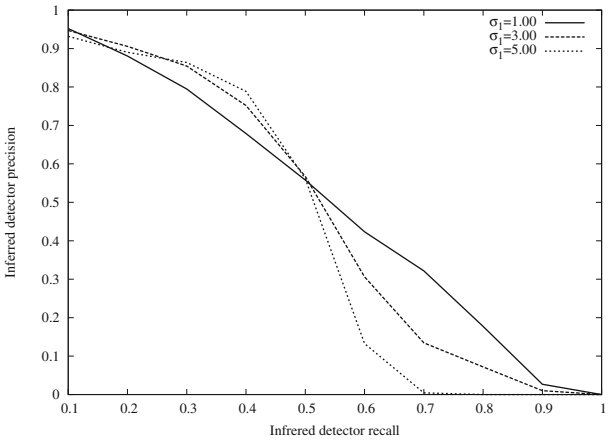


Fig. 11 Inferred precision/recall graph with three different standard deviations for the positive class (mm101 vocabulary, $\mu_0 = 0.0$, $\sigma_0 = 1.0$, $\mu_1 = 3.0$)

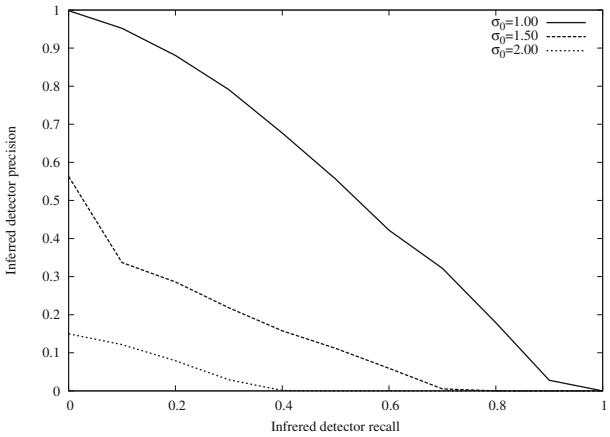


Fig. 12 Inferred precision/recall graph with three different standard deviations for the negative class (mm101 vocabulary, $\mu_0 = 0.0$, $\mu_1 = 3.0$,$\sigma_1 = 1.0$)

shots in the negative class than in the positive class. Figure 13 shows the effect of this scenario for a single concept. Let us assume 1,000 shots in the positive class and 100,000 in the negative class. For the positive class in Fig. 13, we expect 50% of the shots to have a confidence score of $o > 3$, which are 5,000 shots, because of the cumulative distribution function of a Gaussian. For the negative class with standard deviation $\sigma_0 = 1.5$ we expect 2.2% of the shots, which are roughly 2,200 shots, to have a confidence score $o > 3$. If we increase the standard deviation of the negative class to $\sigma_0 = 2.0$, the expected percentage of shots with $o > 3$ increases to roughly 6.6% or 6,600 shots. Now, this triplication of negative shots with high confidence scores ($o > 3$) causes the precision to drop at lower recall levels. This effect is strengthened by an increasing ration of shots in the negative class to shots in the positive class.

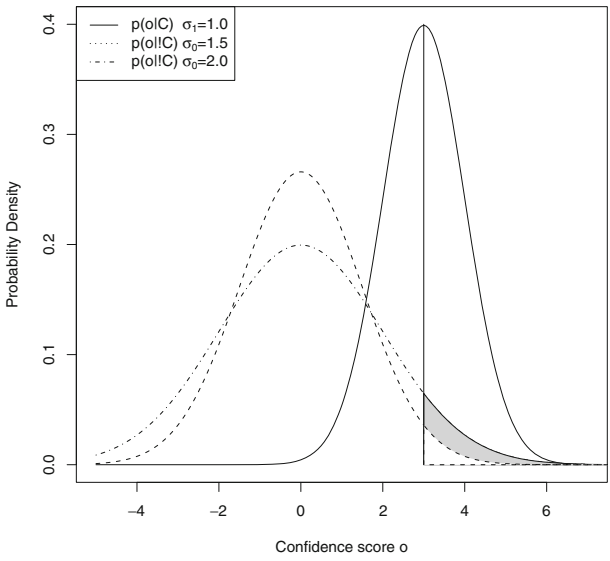


Fig. 13 The effect of changing the standard deviation of the negative class (The gray area reflects the increased percentage of shots of the negative class in with a score higher than $o > 3$)

For the retrieval models which rely on a posterior probability measure (or a classification derived thereof), the number of training examples to fit the sigmoid function is of importance. From Fig. 9 we see that with less than 5,000 samples fitting errors lead to performance decreases. However, beyond 5,000 samples the performance is stable.

Figure 10 shows that the detector MAP is not always strongly correlated with the search performance and the correlation further decreases for a low detector MAP. A likely explanation is that the MAP measure for concept detectors does not capture the extremeness of the confidence scores. A detector with a high standard deviation emits for many shots in the positive class extremely high and low confidence scores. The shots with extremely high confidence scores cause the detector MAP to be high, since the MAP measure favors correctly highly ranked shots. On the other hand, many shots from the negative class will have higher confidence scores than those shots in the positive class with extremely low confidence scores and will therefore be ranked higher in the search rankings. This effect is also shown in Fig. 11.

6 Conclusions and future work

This paper proposed a Monte Carlo Simulation approach to answer the following research questions: (1) *What is the impact of detector performance on the performance of concept-based retrieval engines, and* (2) *will these engines be applicable to real-life search tasks if detector performance improves in the future?* For the prediction we considered the mean average precision (MAP) of the search as a performance measure. We assume that a search performance of 0.20 MAP for a concept-based retrieval engine is sufficient for real-life applications, which is a performance often achieved by retrieval engines of participants of the TREC conference [13].

Detector model The proposed probabilistic model for the Monte Carlo Simulation consists of the parameters of two Gaussian distributions, a mean and a standard deviation for the positive and the negative class. This model was supported by empirical evidence of real detectors and related work [11] of general classifiers. We used equal parameter settings for all detectors, assuming detectors of similar discriminative power. While this is clearly not realistic, it allowed us to focus on the influence of the general detector performance. We step-wise modified the parameters of the model which allowed us to predict the expected search performance of retrieval engines for improving detector performance.

Simulation setup The experiments were carried out on the TRECVid 2005 and TRECVid 2007 development collections, where relevance and concept occurrences were known. We used three concept vocabularies: (1) the MediaMill vocabulary consisting of 101 concepts; (2) the Vireo vocabulary consisting of 374 concepts and (3) the concept vocabulary which resulted from a joint annotation effort during TRECVid 2007–2009 which consisted of 61 concepts. Furthermore, we investigated the influence of a concept selection and weighting method by comparing the following alternatives. First, we used an oracle concept selection and weighting method, which selected the concepts and their weights in hindsight, assuming the knowledge of the documents' relevance. Second, we used our previously proposed Annotation-Driven Concept Selection method [3] which realistically selected concepts. The detector

scores were produced by our open source software detectsim[4] which simplifies the generation of reproducible detector simulations.

Video shot retrieval We investigated the change in search performance of four retrieval models when varying three different parameters of the detector model. The influence of each change is concluded in the following.

When increasing the mean of the positive class, we found that the two retrieval models based on concept-based document representations, the Binary Independence Model (BIM) [20] and Probabilistic Framework for Unobservable Events (PRFUBE) [2]. However, the search performance of BIM increases more slowly due to a high misclassification rate with low detector performance. The Borda–Count model [10], which is based on the ranks of confidence scores, first showed similar performance as BIM but reaches a lower search MAP. The Pointwise Mutual Information Weighting Scheme model [36] has a lower performance than the other models. PRFUBE is the first to achieve real-life sufficient performance under realistic weight settings with an approximate detector performance of 0.60 MAP, which is still far from perfect classification. The BIM model achieved the search performance of 0.20 MAP at a much higher detector performance of 0.88 MAP. Given our assumptions, we therefore conclude that retrieval models using concept-based document representations will be applicable to real-life applications once concept detectors reached a high performance level of 0.60 MAP.

The increase of the standard deviation of the positive class distinctly reduced the search performance of all retrieval models while detector performance was only affected slightly. From this we conclude that current retrieval models are sensitive to a higher variability of confidence scores in the positive class. An increase of the standard deviation of the negative class, resulted in a more distinct search performance reduction. The detector performance was also distinctly reduced. This effect was attributed to the fact that there are usually many more shots in the negative class and a higher standard deviation of this class resulted in many highly ranked false positives. We conclude that both, search and detector performance, are sensitive to changes in the standard deviation of the positive and the negative class.

Furthermore, we investigated the influence of fitting errors of the posterior probability function because of limited training examples. The Borda–Count model was unaffected because it only depends on the ranks of confidence scores. All other retrieval models showed decreased performance with less than 5,000 training examples.

Predictive power of detector MAP We also found that the MAP performance measure for concept detectors is not always a good indicator of the search performance. The increase of the standard deviation of the positive class caused a severe search performance decrease while the detector performance reduced only slightly.

Future work In this paper, we focused on modeling the influence of the independent distribution of confidence scores of concept detectors which is arguably their most important characteristic. Therefore, in the future we plan to further improve the

[4] http://detectsim.sourceforge.net/

model's fit with reality by including dependencies among the confidence scores. Furthermore, we will investigate other measures for the detector performance which consider the overlaps of the distribution of confidence scores in the positive and negative class, such as the Kullback Leibner Divergence [4].

Concluding remark The simulation approach proposed in this paper can be used to evaluate retrieval models without the (immediate) need of real detector outputs. Since building concept detectors is a challenging task, this lowers the entry costs for new researchers interested in concept-based retrieval. Furthermore, the simulation allows predicting the development of the search performance of existing retrieval models under varying detector performance.

Acknowledgements This research was funded by the CTIT strategic research orientation Natural Interaction in Computer-mediated Environments (SRO-NICE).[5] We want to thank the anonymous reviewers for their valuable feedback.

References

1. Aly R, Hiemstra D (2009) A simulator for concept detector output. Technical Report TR-CTIT-09-40, University Twente, Enschede. http://eprints.eemcs.utwente.nl/16544/
2. Aly R, Hiemstra D, de Vries AP, de Jong F (2008) A probabilistic ranking framework using unobservable binary events for video search. In: CIVR '08: proceedings of the international conference on content-based image and video retrieval. ACM, pp 349–358. ISBN 978-1-60558-070-8. doi:10.1145/1386352.1386398
3. Aly R, Hiemstra D, de Vries AP (2009) Reusing annotation labor for concept selection. In CIVR '09: proceedings of the international conference on content-based image and video retrieval, ACM. ISBN 978-1-60558-070-8
4. Arndt C (2001) Information measures: information and its description in science and engineering, Springer
5. Ayache S, Quénot G (2007) Evaluation of active learning strategies for video indexing. Signal Process Image Commun 22(7–8):692–704
6. Bather J (2000) Decision theory. An introduction to dynamic programming and sequential decisions. Wiley-interscience series in systems and optimisation. Wiley, West Sussex, England
7. Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
8. Christel MG, Hauptmann AG (2005) The use and utility of high-level semantic features in video retrieval. In: Image and video retrieval, vol 3568/2005. Springer, Berlin / Heidelberg, pp 134–144. ISBN 978-3-540-27858-0. doi:10.1007/1152634617. http://www.springerlink.com/content/1mf31v38vgltkg1r/

[5] http://www.ctit.utwente.nl/research/sro/nice/

9. Croft WB, Harding S, Taghva K, Borsack J (1992) An evaluation of information retrieval accuracy with simulated ocr output. In: In proceedings of the third annual symposium on document analysis and information retrieval, pp 115–126
10. Donald KM, Smeaton AF (2005) A comparison of score, rank and probability-based fusion methods for video shot retrieval. In: Image and video retrieval, vol 3568/2005. Springer, Berlin / Heidelberg, pp 61–70. ISBN 978-3-540-27858-0. doi:10.1007/1152634610. http://www.springerlink.com/content/9jwatefm7p00dmkm/
11. Hastie T, Tibshirani R (1996) Classification by pairwise coupling. Technical report, Standford University and University of Torronto
12. Hauptmann AG, Yan R, Lin W-H, Christel M, Wactlar H (2007) Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news. IEEE Trans Multimedia 9–5:958–966. doi:10.1109/TMM.2007.900150
13. Hawking D (2000) Overview of the trec-9 web track. In: Voorhees EM, Harman DK (eds) NIST special publication 500-249: the ninth text retrieval conference (TREC 9), p 87
14. Jiang Y-G, Yang J, Ngo C-W, Hauptmann A (2010) Representations of keypoint-based semantic concept detection: a comprehensive study. IEEE Trans Multimedia 12(1):42–53. ISSN 1520-9210. doi:10.1109/TMM.2009.2036235
15. Lin H-T, Lin C-J, Weng R C (2007) A note on platt's probabilistic outputs for support vector machines. Mach Learn 68(3):267–276. ISSN 0885-6125 (Print) 1573-0565 (Online). doi:10.1007/s10994-007-5018-6. http://www.springerlink.com/content/8417v9235m561471/
16. Metropolis N, Ulam S (1949) The monte carlo method. J Am Stat Assoc 44(247):335–341. ISSN 01621459. http://www.jstor.org/stable/2280232
17. Naphade M, Smith J, Tesic J, Chang S-F, Hsu W, Kennedy L, Hauptmann AG, Curtis J (2006) Large-scale concept ontology for multimedia. IEEE Multimedia 13(3):86–91. ISSN 1070-986X. doi:10.1109/MMUL.2006.63
18. Platt J (2000) Advances in large margin classifiers, chapter probabilistic outputs for support vector machines and comparison to regularized likelihood methods. MIT Press, Cambridge, MA, pp 61–74
19. Press W, Teukolsky S, Vetterling W, Flannery B (1992) Numerical recipes in C, the art of scientific computing, 2nd edn. Cambridge University Press
20. Robertson SE, van Rijsbergen CJ, Porter MF (1981) Probabilistic models of indexing and searching. In: SIGIR '80: proceedings of the 3rd annual ACM conference on research and development in information retrieval. Butterworth & Co, Kent, UK, pp 35–56. ISBN 0-408-10775-8
21. Ross SM (2006) Introduction to probability models. Academic Press. ISBN 0125980620.
22. Sangswang A, Nwankpa C (2003) Justification of a stochastic model for a dc-dc boost converter. In: Industrial electronics society, 2003. IECON '03. The 29th annual conference of the IEEE, vol 2, pp 1870–1875. doi:10.1109/IECON.2003.1280345
23. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and trecvid. In: MIR '06: proceedings of the 8th ACM international workshop on multimedia information retrieval. ACM Press, New York, NY, USA, pp 321–330. ISBN 1-59593-495-2. doi:10.1145/1178677.1178722
24. Smeaton AF, Over P, Kraaij W (2008) High level feature detection from video in TRECVid: a 5-year retrospective of achievements. In: Divakaran A (ed) Multimedia content analysis, theory and applications, Springer
25. Snoek CGM, Worring M (2007) Are concept detector lexicons effective for video search? In: 2007 IEEE international conference on multimedia and expo, pp 1966–1969. doi:10.1109/ICME.2007.4285063
26. Snoek CGM, Worring M (2009) Concept-based video retrieval. Found Trends Inf Retr 4(2):215–322
27. Snoek CGM, Worring M, van Gemert JC, Geusebroek J-M, Smeulders AWM (2006) The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Multimedia '06: proceedings of the 14th annual ACM international conference on multimedia. ACM Press, New York, NY, USA, pp 421–430. ISBN 1-59593-447-2. doi:10.1145/1180639.1180727
28. Snoek CGM, van de Sande K, de Rooij O, Huurnink B, van Gemert J, Uijlings J, He J, Li X, Everts I, Nedovic V, van Liempt M, van Balen R, de Rijke M, Geusebroek J, Gevers T, Worring M, Smeulders A, Koelma D, Yan F, Tahir M, Mikolajczyk K, Kittler J (2009) The mediamill TRECVid 2009 semantic video search engine. In: Proceedings of the 9th TRECVid workshop, Gaithersburg, USA
29. Taylor JR (1996) An introduction to error analysis, 2 edn. University Science Books. ISBN 093570275X

30. Toharia P, Robles OD, Smeaton AF, Rodríguez A (2009) Measuring the influence of concept detection on video retrieval. In: CAIP 2009 - 13th international conference on computer analysis of images and patterns, Springer
31. Witbrock M, Hauptmann AG (1997) Speech recognition and information retrieval: experiments in retrieving spoken documents. In: In proceedings of the the DARAP speech recognition workshop 1997, pp 2–5
32. Yan R (2006) Probabilistic models for combining diverse knowledge sources in multimedia retrieval. PhD thesis, Canegie Mellon University
33. Yan R, Hauptmann AG (2003) The combination limit in multimedia retrieval. In: Multimedia '03: proceedings of the eleventh ACM international conference on multimedia. ACM, New York, NY, USA, pp 339–342. ISBN 1-58113-722-2. doi:10.1145/957013.957086
34. Yan R, Hauptmann AG (2007) A review of text and image retrieval approaches for broadcast news video. Inf Retr 10(4–5):445–484. ISSN 1386-4564 (Print) 1573-7659 (Online). doi:10.1007/s10791-007-9031-y. http://www.springerlink.com/content/r742245481q23631/
35. Yang J, Hauptmann AG (2008) (Un)reliability of video concept detection. In: CIVR '08: proceedings of the 2008 international conference on content-based image and video retrieval. ACM, New York, NY, USA, pp 85–94. ISBN 978-1-60558-070-8. doi:10.1145/1386352.1386367
36. Zheng W, Li J, S, Z, Lin F, Zhang B (2006) Using high-level semantic features in video retrieval. In: Image and video retrieval, vol 4071/2006. Springer, Berlin / Heidelberg, pp 370–379. ISBN 978-3-540-36018-6. doi:10.1007/11788034_38

Robin Aly recently received his PhD from the University of Twente in the Netherlands. His thesis is entitled Modeling Representation Uncertainty in Concept-based Multimedia Retrieval. He contributed to over 15 research papers. His research interests include formal models of information retrieval, modeling uncertainty in information retrieval, and multimedia retrieval. He was involved in the organization of the Dutch Belgium Information Retrieval workshop 2009 and in the program committee of several international conferences.

Djoerd Hiemstra is associate professor at the database group of the University of Twente in the Netherlands. He wrote an often cited PhD thesis on language models for information retrieval and contributed to over 140 research papers in the field of information retrieval. His research interests include formal models of information retrieval, multimedia retrieval, federated retrieval, and XML retrieval. He was involved in the local organization of the ACM SIGIR 2007 conference in Amsterdam, and in the organization of several workshops including editions of the Dutch–Belgian Information Retrieval workshop series. Djoerd has been involved in various national and international research projects, and was awarded a prestigious Netherlands Organization for Scientific Research (NWO) Vidi grant. For more information, see: http://www.cs.utwente.nl/~hiemstra/

Franciska de Jong is full professor of language technology at the University of Twente since 1992. She is also affiliated to the Erasmus University in Rotterdam, where she is managing director of the Erasmus Studio. She studied Dutch language and literature at the university of Utrecht, did a PhD track in theoretical linguistics and started to work on language technology in 1985 at Philips Research where she worked on machine translation. Currently, her main research interest is in the field of multimedia indexing, text mining, semantic access, cross-language retrieval and the disclosure of cultural heritage collections (in particular spoken audio archives), and she coordinates a research programme in this area within the Human Media Interaction group. She is frequently involved in international programme committees, expert groups and review panels, and has initiated a number of EU-projects. In 2001–2003 she was a member of the EU/NSF 'spoken word archives' working group. She is project leader of the MultimediaN-project on semantic multimedia access (2004–2009), principal investigator of the NWO-CATCH project CHoral (2006–2010) and coordinator of IST project PuppyIR (2009–2012). Since 2008 she is a member of the Governing Board of the Netherlands Organization for Scientific Research (NWO).

Peter M. G. Apers received his MSc and PhD from the Free University of Amsterdam. After receiving his MSc he spent one year at UC Santa Cruz and after his PhD half a year at Stanford University. He is currently a faculty member of the Computer Science Department of the University of Twente. He is also scientific director of CTIT, the ICT institute of the University of Twente, and of NIRICT, the ICT institute of the three Dutch universities of technology. He served as (vice) program chair of major conferences such as VLDB, ICDE, and EDBT. Furthermore, he has been Editor-in-Chief of the VLDB Journal. His research interest are database querying and multimedia information retrieval.