Concept Detectors: How Good is Good Enough?

A Monte Carlo Simulation Based Approach

Robin Aly
University Twente
P.O. Box 217,
7500AE Enschede, The Netherlands
r.aly@ewi.utwente.nl

Djoerd Hiemstra
University Twente
P.O. Box 217,
7500AE Enschede, The Netherlands
hiemstra@ewi.utwente.nl

ABSTRACT

Today, semantic concept based video retrieval systems often show insufficient performance for real-life applications. Clearly, a big share of the reason is the lacking performance of the detectors of these concepts. While concept detectors are on their endeavor to improve, following important questions need to be addressed: "How good do detectors need to be to produce usable search systems?" and "How does the detector performance influence different concept combination methods?". We use Monte Carlo Simulations to provide answers to the above questions. The main contribution of this paper is a probabilistic model of detectors which outputs confidence scores to indicate the likelihood of a concept to occur. This score is also converted into a posterior probability and a binary classification. We investigate the influence of changes to the model's parameters on the performance of multiple concept combination methods. Current web search engines produce a mean average precision (MAP) of around 0.20. Our simulation reveals that the best performing video search method achieve this performance using detectors with 0.60 MAP and is therefore usable in real-life. Furthermore, perfect detection allows the best performing combination method to produce 0.39 search MAP in a artificial environment with Oracle settings. We also find that MAP is not necessarily a good evaluation measure for concept detectors since it is not always correlated with search performance.

1. INTRODUCTION

Content based video retrieval currently mainly focuses on the improvement of detectors of semantic concepts in video shots. Here, semantic concepts are events of which a user can judge the occurrence or absence, for example *Outdoor* and *Singing*. Semantic concepts are also often referred to as High Level Features [18, 7], because of their meaning to humans, and as visual concepts [8], because of their predominate occurrence in the visual part of the video. On the other hand, there is research on combining the output of the concept detectors to answer the information needs of users. This search strategy is promising since it attempts to separate the concept based retrieval problem from the detection of the concepts. However, currently the performance of such retrieval systems often prohibits their application in reality. Clearly, the performance of the overall system heavily depends on the performance of the detectors. However, we also need to answer the questions "How good do the detectors need to be to generate a satisfying overall search performance?" And "What is the effect of the detector performance on different proposed combination methods?" This paper investigates the application of a Monte Carlo Simulation based approach to answer these questions by simulating detectors using a probabilistic model of them.

Hauptmann et al. [7] were the first to use a simulation based approach to predict achievable performance of concept based video retrieval systems. There, 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 occurrence and absence. While the model of a binary classifier was useful to study the general applicability of concepts for search, most retrieval systems today employ confidence scores or a probability measure based on this score. 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 *Barack Obama* is clearly useful for the query "Talks from the current U.S. President". However, the corresponding detector might never positively classify a shot for the concept *Barack Obama* but may find few shots more likely to contain *Barack Obama* than others, which could be exploited. Therefore, the simulation approach in this paper generates confidence scores for each shot and concept in a lexicon which then can be transformed into probability measures as well as classifications.

Our approach follows the Monte Carlo Method [10] to estimate the detector and search performance of a retrieval system given certain detector characteristics. For the simulation we define a probabilistic model of detectors: First, we assume that detectors are independent from each other and that each detector emits for each shot a single, real valued, confidence score. Second, we make the assumption that these confidence scores are normally distributed in the set of shots where the concept occurs and likewise where it does not (the positive and negative class). This assumption is supported by studies of actual detector outputs in this paper and Hastie in [22]. Based on this model and a col-

lection with known concept occurrences we generate a set of randomized detector scores on which we execute different search runs using different state-of-the-art combination methods. To rule out random effects we repeat the process multiple times to calculate the expected mean average precision (MAP), defined for example in [7], of each combination method using the given parameters. We then gradually change the parameters of the model and therefore the performance level of the detectors to investigate the effect of the changed detector characteristics to the overall search performance.

This paper is structured as follows: In Section 2 we give an overview of related work in this area. Section 3 describes our approach to assess the necessary detection performance through simulation. In Section 4, we investigate the results of the simulation on a collection with concept annotations and relevance judgments. Section 5 concludes this paper and proposes future work.

2. RELATED WORK

In this section we review related work. This paper proposes a simulation approach based on Monte Carlo Simulations [10]. In literature, the term Monte Carlo Simulation is used for a wide variety of different methods. Here, we use it for a general procedure to calculate the expected outcome of a deterministic calculation based on a probabilistic model of the inputs. The procedure can be described in the following steps: (1) The definition of a probabilistic model of the inputs to the simulation, the confidence scores and their distribution based on their class in our case. (2) Random generation of a concrete set of inputs using the model, a set of concrete confidence scores in this paper. (3) Execution of a deterministic computation using the generated inputs, in our case the execution of the search and its performance evaluation and (4) Repetition of (2) and (3) to produce multiple results. Finally, (5) average the results of the individual computations into the final result. This calculation is guaranteed to converge with increasing number of repetitions to the expected outcome based on the model.

The majority of current concept detectors are using Support Vector Machines (SVM) [20, 26]. 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 which should separate positive and negative instances. 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 [13]:

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

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 seen for example in [23] and from the fact that the concept detectors are currently evaluated using MAP [18] - a rank based method. 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 proposes in [22] 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 for the positive class. Platt proposes in [13] 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. [9], which is also used in many SVM implementation, for example libSVM [6].

Hauptmann et al. [7] were the first to use a simulation based approach to investigate achievable concept based search performance. In their work, a detector is assumed to be a binary classifier. As a search method they use a linear combination of concept occurrences: $score(s_i) = \sum_j \lambda_j f_{i,j}$. Here, $score(s_i)$ is the retrieval score of shot s_i, λ_j is a concept specific coefficient and $f_{i,j}$ is the state of concept j in shot s_i. The coefficients λ_i are separately set for each query. The coefficient setting which optimize the average precision is found by solving a bounded constrained global optimization problem [24]. The retrieval performance with realistically set coefficients is assumed to achieve 50% of the performance with optimal settings. Concept labels of shots are randomly flipped until the precision recall break even point is reached. We argue that this approach can be improved since the current retrieval systems most often use confidence scores and an uniform break even precision-recall point assumes the same performance among all detectors which is clearly unrealistic.

Furthermore, Hauptmann et al. [7] find that the Mutual Information from Information Theory [3] between a concept and relevance to be the best weight for the influence of a concept. We will use this finding to select the most influential concepts for a query and for setting their combination weights. However, perfect estimation of the weights is infeasible in reality. Therefore, we also employ a concept selection method which can be employed in real systems. There are several works on concept selection, see [12] for an overview. Here, we use a recently proposed method by Aly et al. [1] which uses textual descriptions of an annotated collection to estimate the probability of a concept occurring in relevant shots. Together with an estimate of the probability of encountering a relevant shot the Mutual Information of a concept and relevance can be calculated from this probability. We use this concept selection method since it also provides an estimate for the occurrence probability of a concept in relevant shots - which is used as a parameter in multiple concept combination methods.

We chose following concept combination methods to study

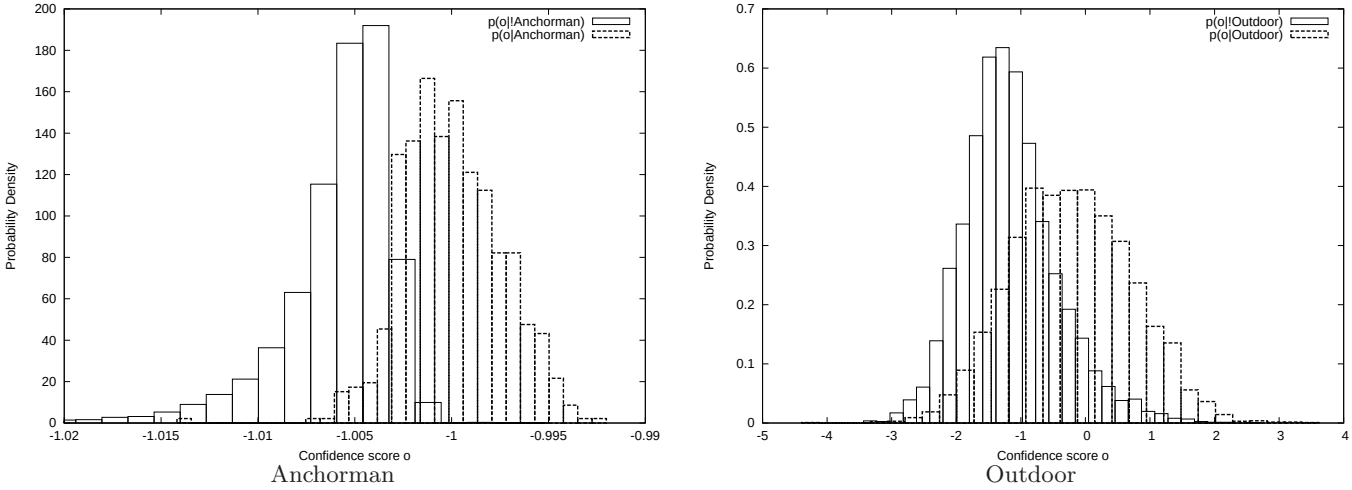


Figure 1: Distributions of Confidence Scores

their behavior under changed detector characteristics: The first method is Borda Count Fusion [4], which is often called reranking. See for example Snoek et al. [21] for its application to video retrieval. After a concept selection method identified influential concepts their detector scores are used in a voting scheme. A shot is ranked the higher it occurs in the rankings of the concepts. The concepts can be assigned weights to emphasize their importance to the query. Zheng et al. introduce in [27] a method which ranks a shot by the product of the point-wise Mutual Information gain of a concept occurrence and the probability that the concept is present in the current shot. Since the authors did not give a name to the method, we refer to this method as the Entropy method. The Probabilistic Retrieval Framework for Uncertain Binary Events (PRFUBE) [2] considers for each concept both possible cases of occurrence and absence in the scoring function to calculate the probability of relevance. In case the detectors are used as binary classifiers the Binary Independence Model (BIM) [15] can be applied where the detected concept occurrences are treated as term occurrence in the BIM model. Since the PRFUBE and BIM method are derived from the Probability Ranking Principle [16] they guarantee optimal performance under some independence assumption. The Borda Count Fusion does only depend on the rank of the confidence score in an ordered list of shots while Entropy, BIM and PRFUBE also depend on the quality of the posterior probability.

Yan proposes in [23] a probabilistic concept combination method based on the confidence scores which is based on logistic regression. It relies on training data for so-called query classes which are used to decompose the query. The training data consists of relevance judgments to example queries and confidence scores. While this is one of the most recent works on concept combination we do not include it into our study since the available training data for the method overlaps with our search collection, the TRECVID 2005 development collection, and therefore would result in over fitting.

In this paper we will assess the effects of the different detector performances on the above combination methods. Note that the Entropy, PRFUBE and BIM combination methods have an optimal "Oracle" weight setting which can

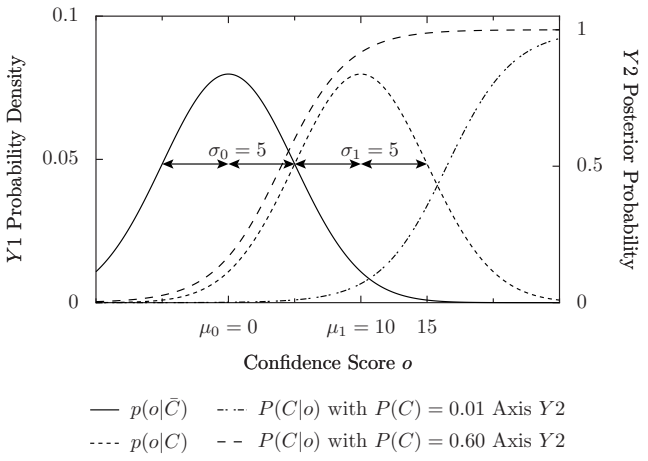


Figure 2: Detector Model

be easily determined by counting shots using relevance judgments and concept occurrence information. The Mutual Information can be determined likewise. For the Borda Count Fusion we assume that the Mutual Information as a weight for each concept to also deliver a kind of perfect weight setting.

3. DETECTOR SIMULATION

In this section we describe the simulation process of the concept detectors. In Section 3.1 we describe our model of an SVM based concept detector. Section 3.2 describes how we calculate from the confidence score a posterior probability. Finally, Section 3.3 explains how the actual randomization in the simulation is performed.

3.1 Detector Model

In this section we describe our probabilistic model of a detector, which is later used for the randomization of confidence scores. Figure 1 shows the confidence score histograms of the concepts *Anchorman* and *Outdoor* for both classes from a base line detector set, described in [20]. The differ-

ent score ranges and the resulting probability density magnitudes are caused by the detector's ability to discriminate between positive and negative examples. We see that both densities have roughly a Gaussian shape. This shape was also proposed by Hastie [22]. A χ^2 goodness of fit test revealed that 31 of the 101 detectors in the lexicon could be accepted as gaussian at a significance level of 0.05. Most of them were concepts with many training examples which suggests that the gaussian shape would also become evident for other concepts, once we had more training examples available for them. Furthermore, Sangswan and Nwankpa argue in [17] that a non-perfect fitting shape of a model only increases the variance of the Monte Carlo estimator, but still allows a trustworthy simulation outcome.

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 simplify the model, we assume that all concepts share the same mean μ_1 and standard deviation σ_1 for the positive and μ_0 and σ_0 for the negative class respectively. Note, that this assumption is strong and certainly does not hold in reality. However, because we focus here on the principle behavior of the detectors 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 an important aspect for building detectors, we argue that the intersection of the areas under the probability density curves has a much higher influence on performance than the absolute ranges of the confidence scores. Clearly, the more the area of the intersection decreases the better the detector is. Our model can adequately simulate this effect by either moving the means apart or by varying the deviation of the classes.

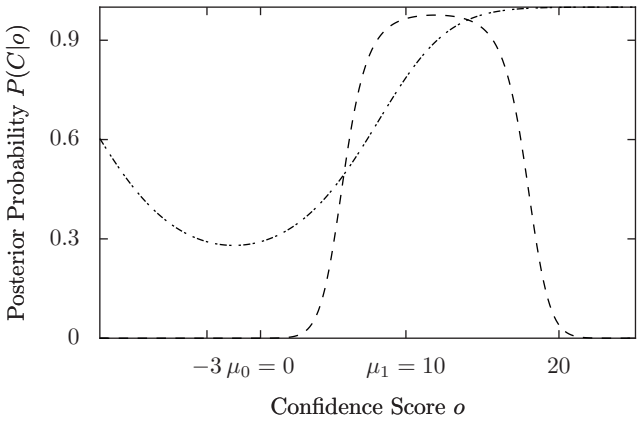
Figure 2 shows the model of a single detector. We also plot the posterior probability of observing the concept given the confidence score using two different priors, one of $P(C) = 0.01$ and one for $P(C) = 0.50$. Considering a confidence score of 15 the posterior probability for a concept with the prior of 0.50 is close to certainty 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 in [7].

3.2 Posterior Probability

As noted by Platt [13], the assumption of two Gaussians for the negative and positive class can lead to unwanted effects, namely a non-monotonic posterior probability function. Figure 3 shows the posterior probability functions of two hypothetical concept detectors defined by the standard formula for posteriors:

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

We see that with a standard deviation of $\sigma_1 = 15$, the posterior probability increases for a confidence score smaller $o = -3$. 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 SVMs (where the classes should be linearly separable). To prevent this effect



$P(C|o)$ with $P(C) = 0.60$ and $\sigma_0 = 5, \sigma_1 = 15$

$P(C|o)$ with $P(C) = 0.60$ and $\sigma_0 = 5, \sigma_1 = 2$

Figure 3: Non-monotonic Posteriors

we use an improved version of the algorithm from Platt [13], suggested by Lin et al. [9], to fit a sigmoid function to the confidence scores. The sigmoid function is defined by the two parameters A and B:

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

Combination methods which depend on the probabilistic output of the concept detectors therefore 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 samples for the fitting process 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.

3.3 Simulation Process

In this section we discuss the actual simulation process which is defined 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 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 the dataset. We then generate confidence scores for the positive and negative class using the prior probability and S samples. Now we use the algorithm described by Lin et al. [9] to fit the sigmoid posterior probability function to the generated samples. After the determination of the sigmoid parameters we iterate over all shots in the annotated collection. For each shot we determine for each concept in the lexicon 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 combination methods 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 [5].

After the randomization, we determine the detector per-

Data: Annotated Collection **C**, Lexicon **L**
Input: $N, S, \mu_0, \sigma_0, \mu_1, \sigma_1$
Result: Randomized collection

// Randomize Prior Estimate
foreach *Concept C in Lexicon* ***L*** **do**
  Calculate $P(C)$ from annotations in **C**
  generate $\lceil S\,P(C) \rceil$ positive samples from $N(\mu_1, \sigma_1)$
  generate $S - \lceil S\,P(C) \rceil$ negative samples $N(\mu_0, \sigma_0)$
  determine A_C and B_C according to [9]
end
// Randomize Detection Output
for *Repetition* $i \in [1..N]$ **do**
  foreach *Shot s in Collection* ***C*** **do**
    foreach *Concept C in Lexicon* ***L*** **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 [13]
      $P(C|o) = \frac{1}{1+exp(A_C o + B_C)}$
      // Transform to Binary Value
      if $P(C|o) > 0.5$ **then**
        $C = 1$
      else
        $C = 0$
      end
    end
  end
  Calculate Detector Performance $DMAP_i$
  Run Search with Combination Method
  Calculate Search Performance $SMAP_i$
end
Report $DMAP = \frac{\sum_i DMAP_i}{N}, SMAP = \frac{\sum_i SMAP_i}{N}$

Algorithm 1: Algorithm for a Simulation Run. N: Number of Repetitions, S: Sample size for Sigmoid fitting, $\mu_0, \sigma_0, \mu_1, \sigma_1$: Model parameters

formance MAP of the detector output ($DMAP_i$). We then execute a search run for each combination method 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 N times to rule out random effects and the results are averaged.

4. SIMULATION RESULTS

In the following we describe the results of the simulation runs. Section 4.1 specifies the setup of the simulations. In the following Section 4.2 we describe what parameter changes are investigated in our experiments. Section 4.3 investigates the comparability of the detector model to a set of real detectors. Section 4.4 presents the results of increasing the distance of the two means and Section 4.5 does the same for the increase of standard deviation. The effects of the number S of training samples for the sigmoid fitting on the search performance are described in Section 4.6 and we discuss the results of the simulations in Section 4.7.

4.1 Simulation Setup

We perform our simulation on the TRECVID 2005 development collection since, to the author's knowledge, this is the only suitable dataset where both concept annotations and relevance judgments are available[1]. We use 24 original queries from TRECVID 2005. To prevent over-fitting when performing realistic concept selections we divide the collection according to the MediaMill Challenge setting [20] into the sub-collections mm.dev and mm.test. The statistics for the collections are summarized in Table 1. We use two concept lexicons in our simulation to ensure that our results are not lexicon specific. The MediaMill lexicon [20] comprises 101 concepts while the LSCOM lexicon [11] has 374 concepts.

Table 1: Collection Statistics

Name	Videos	Shots
tv2005.dev	141-277	43907
mm.dev	141-238	30630
mm.test	239-277	13277

We use a Java based (pseudo) random number generator[2] implementation following a standard algorithm described in [14]. For every simulation run we use a new seed for the generator to ensure a high quality of randomness. To reduce random effects in the results we repeat every simulation run $N = 25$ times. The experiments showed that the simulation results did not change anymore after this number of repetitions. We use in the following the common expression 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 2000 as done for the High Level Feature task in TRECVID [19] to maintain comparability to other results. However, this cut-off level sometimes leads to counter intuitive results since some frequent concepts occur more than 2000 times and consequently even a perfect detector would have an average precision of less than 1.0. Therefore, we consider a maximum of 2000 shots containing the concept.

As mentioned earlier we use the four concept combination methods Borda Count Fusion [4], Entropy [27], BIM [15] and PRFUBE [2]. The latter three require the probability of concept occurrence given relevance as a parameter for the combination weight. For Borda Fusion Count we assume that a wait of the Mutual Information is an ideal setting. The mutual information can be calculated using the three parameters $P(C)$, $P(R)$ and $P(C|R)$. We perform one experiment using 'Oracle' weight settings, where we used the concept annotations and relevance judgments, which we determine by counting.

Furthermore, we perform another experiment of a realistic scenario where we use the concept selection method from Aly et al. [1] which is based on an annotated training corpus. To use this estimation method without introducing over fitting effects we use the collection mm.test for weight estimation and later execute the search only on mm.dev. We determined the prior probability from the dataset assumed for $P(R)$ a small constant. We also have to set the number

[1]The relevance judgments were kindly provided by Rong Yan formally at Carnegie Mellon University [25]
[2]http://www.ee.ucl.ac.uk/ mflanaga/java/PsRandom.html

Table 2: Model Coherence Statistics

Measure	Simulation Result	Simulation Max	Real Detectors
Detector MAP	0.13	0.16	0.15
Search MAP	0.06	0.11	0.10

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 lexicon.

4.2 Simulation Parameter Variation

As our goal is to study the influence of the detector behavior 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 kind of variation and the characteristics of the set of detectors resulting from it: (1) We increase the mean of the positive class – further away from the negative class. In reality, this is the case if the low-level features become increasingly discriminative. (2) We increase the standard deviations of the positive class. The effect is that detectors with a higher standard deviation have more extreme results: for many shots the system is nearly certain that the concept occurs while for many other shots the uncertainty rises. Finally, (3) We increase the number training samples for fitting the sigmoid posterior probability function, which investigates the influence of a lower quality of fit, caused by a low number of training examples, on the search performance.

4.3 Model Coherence

In this section we exemplary investigate the coherence of our model with a real detector set described in [20]. We first fit the model parameters to this detector set. We expect that average detector performance is close to the performance of the real detectors. However, the average search performance of the model does not have to be similar to the real detectors search performance, because of the random distribution of relevant shots. On the other hand, the real search performance should also not be too far off from the performance produced by the model.

First, we train detectors for the MediaMill lexicon using the features provided by the Challenge Experiment 1 [20] using the mm.dev collection and then perform the evaluation on the mm.test dataset. Since we are only interested in the influence of the detector performance on the search performance we use the PRFUBE combination with Oracle weights and 10 concepts. Furthermore, we estimate the model parameters from the confidence scores of the real detectors and set this time each mean and deviation individually. We calculate the mean and the deviation for each class $x \in \{0, 1\}$ and concept c by:

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

where 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 $N = 30$ simulation runs. The results of the comparison are shown in Table 2. We see that the average detector performance of the model (0.13) is lower than the one of the real detectors (0.15). However, the maximal performance achieved by the model exceeds the performance of the real detector (0.16). Our investigations revealed the lower average performance was due to a high correlation of the confidence scores among many shots ($\approx$ 2000) in the mm.test collection because they were near duplicates. Since we generate the confidence scores independently our model is not able to capture these dependencies. However, we argue that the inclusion of this correlation in our model is also not desirable since near duplicates can be handled separately and will not be as frequent in other collections. The search performance of our model is also lower compared to the real detectors, which means that relevant shots were distributed in favor for the real detector set. However, 6 simulation runs achieve an equal or higher performance to the real detectors. We conclude that our model is sufficiently realistic to explain a current, realistic retrieval setting, except the handling of near duplicates.

4.4 Change of Mean

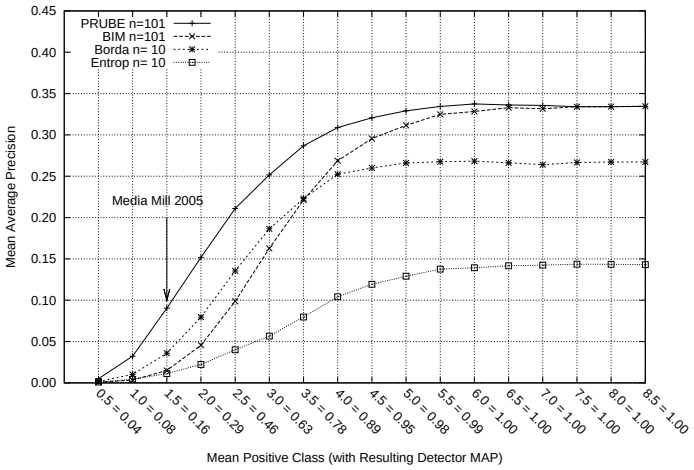
4.4.1 Oracle Combination Weights

Figure 4 (a) and (b) show the simulation results of increasing the mean of the positive class with (a) the MediaMill lexicon and (b) the Vireo lexicon under Oracle combination settings. The Y-Axis shows in all following figures the achieved search MAP of the depicted search runs. The X-Axis shows the mean μ_1 together with the detector MAP which resulted from this setting. The mean of the negative class is kept constant at $\mu_0 = 0$ and therefore is μ_1 also the distance of both means. An increasing distance between the means leads to an increase of the detector MAP. Consequently, the performance of all concept combination methods increases with a growing distance. From a distance of 8.5 the detection can be considered a perfect classification. The Entropy combination method reaches with ten concepts a MAP of 0.15. Borda Count Fusion also performs best when limited to the ten most influential concepts. It achieves an optimal performance of 0.27. The BIM method has a slow start and only reaches a performance of 0.05 MAP at $\mu_1 = 2$ which corresponds to a detector MAP of 0.29. Afterwards, its performance increases faster than the two previously mentioned methods and reaches at $\mu_1 = 8.5$ a MAP of 0.33. PRFUBE consistently shows a better performance than all other combination methods and achieves with $\mu_1 = 8.5$ a search performance of 0.35 MAP. The latter two combination methods performed best with the usage of all concepts in the lexicon.

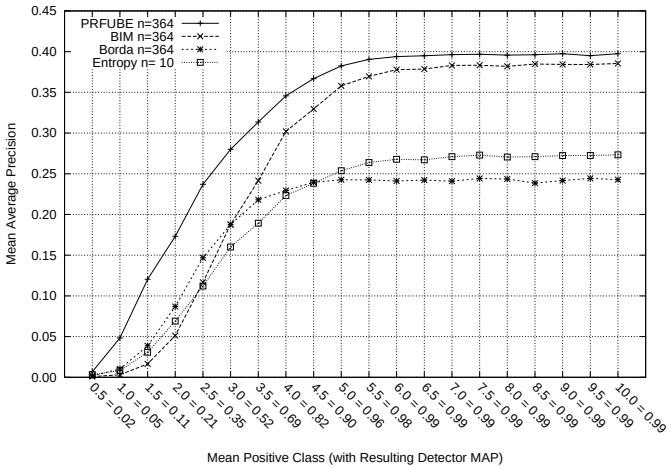
Figure 4 (b) shows the results of simulation runs using the LSCOM lexicon. The results are similar to the usage of the MediaMill lexicon. Notable is that this time the Entropy method achieves a better performance than Borda Count Fusion. The reason is probably the existence of more only positive influential concepts - which can be exploited by the Entropy method. The higher number of concepts allows PRFUBE to increase its performance to 0.39.

4.4.2 Realistic Combination Weights

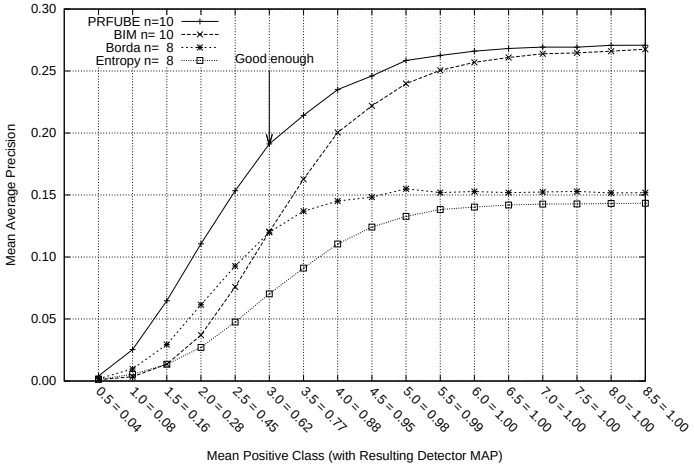
In Figure 4 (c) and (d) the search performance of the combination methods on the mm.dev collection is shown. The combination weights were realistically estimated from the mm.test collection. Sub-figure (c) shows the performance using the MediaMill lexicon. Because the weights are now



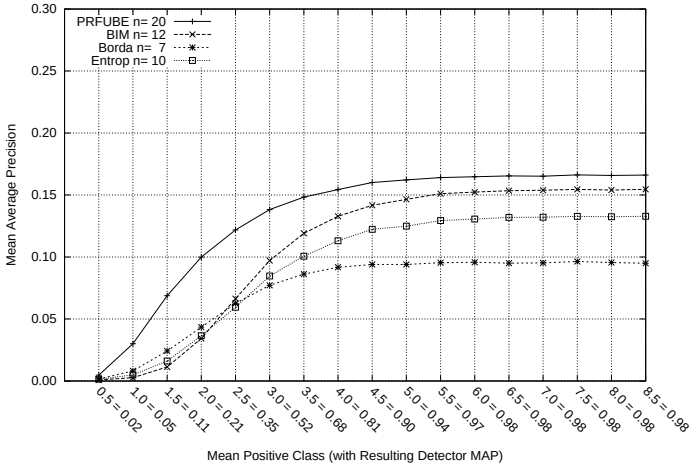
(a) MediaMill Oracle Concept Weights



(b) LSCOM Oracle Concept Weights



(c) MediaMill Real Concept Weights



(d) LSCOM Real Concept Weights

Figure 4: Change of Mean μ_1 $(\mu_0 = 0.0, \sigma_0 = 1.0, \sigma_1 = 1.0)$

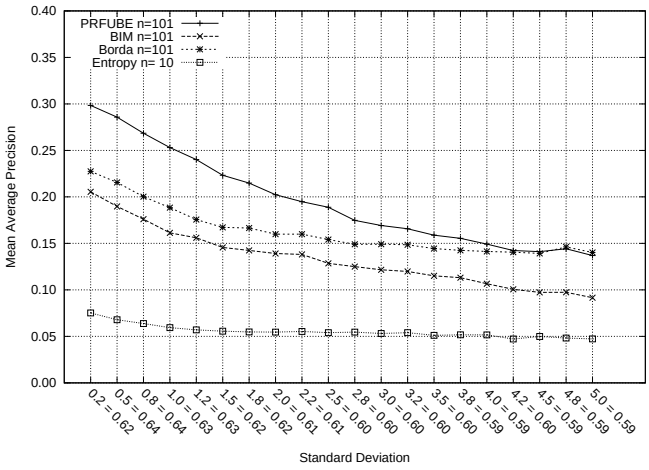
estimated by a realistic concept selection method, the performance is clearly lower. An exception is the Entropy method which stays very close to its Oracle MAP of 0.15. The performance of the search methods relative to each other stays approximately the same. Noticeable is that Borda Count Fusion is not able to leverage its performance gain compared to the Entropy method as in the Oracle setting. Sub-figure (d) shows simulation runs using the LSCOM lexicon. All methods perform worse compared to using the MediaMill lexicon. A likely explanation is that with a growing concept lexicon the chance of selecting bad concepts or wrong weights increases.

4.5 Change of Standard Deviation

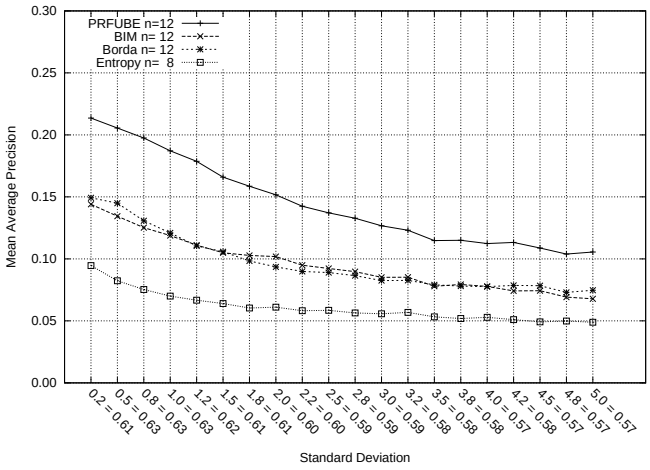
Figure 5 (a) shows the results of a change of the standard deviation of the positive class using Oracle weight settings. We fix all other model parameters as follows: $\mu_0 = 0, \mu_1 = 3, \sigma_0 = 1$. Therefore the performance at $\sigma_1 = 1$ corresponds to the performance of Figure 4 at $\mu_1 = 3$. An increase of the standard deviation of the positive class increases the uncertainty and therefore the difficulty of the search. Consequently, all combination methods show a lower performance with an increasing deviation. After an initial loss of 0.07 MAP the rank based combination method Borda Count Fusion performs equally with the PRFUBE method from $\sigma_1 = 4$ onwards. The Entropy method quickly stabilizes at MAP 0.05. The two methods PRFUBE and BIM show a continuous performance loss.

Sub-figure 5 (b) shows the increase of the standard deviation with weights from a realistic concept selection method. For both Lexicons the PRFUBE method stays around 0.03 above the other combination methods. The Entropy method shows a worse performance than Borda Count Fusion and BIM. It is interesting that in both Sub-figures (a) and (b) the detector MAP only changes by around 5% while the search performance reduces by around 50%. This yields that the MAP of detector scores is not always a good indicator of search performance.

4.6 Sigmoid Fitting



(a) MediaMill Oracle Concept Weights



(b) MediaMill Real Concept Weights

Figure 5: Change of Standard Deviation σ_1 ($\mu_0 = 0.0, \sigma_0 = 1.0, \mu_1 = 3.0$)

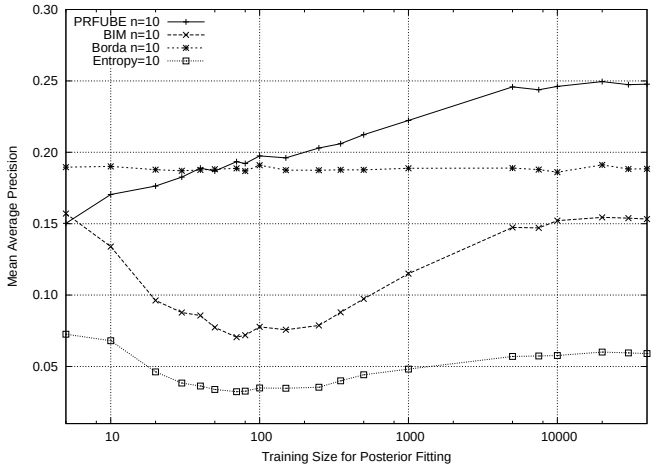


Figure 6: Influence of Training Size S using the MediaMill lexicon / Oracle Concept Weights ($\mu_0 = 0.0, \sigma_0 = 1.0, \mu_1 = 3.0, \sigma_1 = 1.0$)

Figure 6 shows the results of an increasing number of training samples S used in the fitting procedure for the posterior probability function. Here we used the MediaMill lexicon together with Oracle concept weight settings. The X-Axis shows the training size S on a log-scale since smaller training sizes are of higher interest. Except of small random effects, the Borda Count Fusion method shows constant performance since it does not depend on the probabilistic output.

For the BIM and Entropy combination method the search performance decreases until a sample size of $S = 100$. The reason is that in very small training samples of $S = 5$ the minimum number of one positive example drastically over represents the positive class. Therefore, the posterior probabilities are strongly biased towards higher values and the true positive classifications rise. Since the false negatives are the biggest problem for the BIM method its performance rises. The same holds for the Entropy method because the ranking formula only considers the probability of concept occurrence in the shot. With an increasing training set size this effect diminishes. However, the fitting errors due to the still relatively small sample persist. This causes the posteriors to be randomly too high or too low which decreases the performance of both methods. Beyond a minimum performance the performance rises again due to a more representative and bigger sample. The performance of the BIM and Entropy method stabilizes after $S = 5000$ because of increasingly accurate estimates of the parameters for the sigmoid function.

The PRFUBE method improves linearly from 0.15 search MAP using 5 training samples to 0.24 search MAP with 5000 samples. Beyond 5000 samples it stays approximately constant. It is the most affected by "upwards" biased posterior probabilities with a small sample size. However, its performance then increase despite random estimation errors of the sigmoid parameters at sample sizes around 100.

4.7 Discussion

The question of "How good do concept detectors have to be?" can be best answered from Figure 4 (c) and (d) since we want to evaluate the applicability of the search system in real-life. We see that for the MediaMill lexicon a detector performance of 0.60 MAP results in a search performance of the best performing combination method PRFUBE of 0.20 MAP, which is similar to todays web search engines. For the LSCOM lexicon, this performance is never reached due to the difficulty of selecting good concepts from a bigger lexicon. Furthermore, Figure 5 shows that it is important to keep the deviations of the positive and negative class small, since otherwise a high detector MAP will not positively influence the search performance as much.

The PRFUBE combination method consistently performs best while the BIM method achieves approximately the same performance after a slow start. The Borda Count Fusion is often better than the BIM methods in lower performance regions but stabilizes at a lower performance level when the detectors approach certainty. The Entropy method 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 Fusion.

For combination methods which rely on a posterior probability (or on a derived classification thereof) the number of training examples to fit the sigmoid function is of importance. From Figure 6 we see that with less than 5000 samples fitting errors lead to performance decreases. However, beyond 5000 samples the performance is stable.

The realistic concept combination with the LSCOM lexicon in Sub-figure 4 (d) showed that the used concept selection method suffers from a too wide choice of concepts. While the performance with the MediaMill lexicon had only a moderate decrease compared to the Oracle settings the LSCOM lexicon decreased by 50%.

An interesting finding in the change of standard deviations is that the increase of the detector MAP is not strongly correlated with the search performance. While the detector performance in Sub-figure 5 (d) only mildly decreases over the displayed interval the search performance decreases by 50%. Therefore, a measure for the "amount of overlap" of the scores from the positive and negative class seems to be more useful.

5. CONCLUSION AND FUTURE WORK

This paper investigated a Monte Carlo Simulation based approach to answer two important questions, which can not be answered with current concept detectors. We first defined a model of a concept detector and used it to generate confidence scores of a given characteristic. We then executed a search run on the generated output and determined the detector and search performance. Multiple repetitions were used to reduce random effects.

For the first question "How good do detectors need to be?" we found that the best today's search method would achieve a MAP of 0.20 using a detector set with an approximate MAP of 0.60. This is a typical performance of current Web search engines and therefore sufficient for real-life application. We conclude that concept detector still need to be improved. However, the focus on the improvement can be shifted away at a performance around 0.60 MAP - which is still far from perfect classification.

On the question "How does the detector performance influence the search performance of individual concept combination methods?", we found that the two retrieval models based on the probability of relevance principle [16], BIM and PRFUBE, can exploit the full concept lexicon under an Oracle combination weight setting. However, the search performance of BIM increases much slower due to a high misclassification rate in the beginning. Borda Count Fusion first showed similar performance as BIM but reaches a lower search MAP. The Entropy method has a lower performance than the other methods. The maximum reached performance is 0.39 search MAP by the PRFUBE and the LSCOM lexicon. Additionally, PRFUBE showed in most experiments the best performance among the four combination methods. Furthermore, we investigated the influence of fitting errors of the posterior probability function. While Borda Count Fusion was unaffected - since it does only depend on the detector's ranking, all other combination methods showed decreased performance beneath 5000 training samples.

We also found that the mean average precision of detectors is not necessarily a good indicator of the search performance. Since the increase of the standard deviation of the positive class causes a sever search performance decrease while the detector performance only changes slightly. Therefore, we plan to investigate other measures which consider the overlap of the score distributions from the positive and negative class, such as the Kullback Leibner Divergence [3], in future work.

In this paper we focused on the main characteristics of a concept detector. However, we plan to add more diversity to our model to further improve its fit with reality in future work. Furthermore, the simulation approach in this paper can be used as a way to evaluate concept combination techniques without the (immediate) need of real detector outputs. This lowers the entry costs for new researchers interested in concept combination methods and enables all researchers to study the behavior of the algorithms using different detector characteristics. Therefore, we also plan to investigate ways to release the output of this framework to the community in the future. Finally, we also plan to apply our detector simulations to other datasets outside the TRECVID framework.

6. ACKNOWLEDGMENTS

We want to thank the anonymous reviewers for their helpful comments and Yahoo! for the support of our research with computer hardware.

7. REFERENCES

[1] Robin Aly, Djoerd Hiemstra, and Arjen P. de Vries. Reusing annotation labor for concept selection. In *CIVR '09: Proceedings of the International Conference on Content-Based Image and Video Retrieval 2009*. ACM, 2009.

[2] Robin Aly, Djoerd Hiemstra, Arjen P. de Vries, and Franciska de Jong. 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 2008*, pages 349–358, 2008.

[3] C. Arndt. *Information Measures: Information and its description in Science and Engineering*. Springer, 2001.

[4] Javed A. Aslam and Mark Montague. Models for metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, New York, NY, USA, 2001. ACM.

[5] J. Bather. *Decision Theory. An Introduction to Dynamic Programming and Sequential Decisions*. Wiley-Interscience Series in Systems and Optimisation. John Wiley and Sons, West Sussex, England, 2000.

[6] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[7] A. Hauptmann, Rong Yan, Wei-Hao Lin, M. Christel, and H. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. In *IEEE Transactions on Multimedia*, volume 9-5, pages 958–966, August 2007.

[8] J.R. Kender and M.R. Naphade. Visual concepts for news story tracking: analyzing and exploiting the nist tresvid video annotation experiment. In *TODO FILL IN*, volume 1, pages 1174–1181, June 2005.

[9] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, October 2007.

[10] Nicholas Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[11] Milind Naphade, John R. Smith, Jelena Tesic, Shih-Fu Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[12] Apostol (Paul) Natsev, Alexander Haubold, Jelena Tešić, Lexing Xie, and Rong Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 991–1000, New York, NY, USA, 2007. ACM.

[13] J. Platt. *Advances in Large Margin Classifiers*, chapter Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, pages 61–74. MIT Press, Cambridge, MA, 2000.

[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press use, 2nd edition, 1992.

[15] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56, Kent, UK, 1981. Butterworth & Co.

[16] S.E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33:294–304, 1977.

[17] A. Sangswang and C.O. Nwankpa. 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*, volume 2, pages 1870–1875 Vol.2, Nov. 2003.

[18] A.F. Smeaton, P. Over, and W. Kraaij. High level feature detection from video in TRECVid: a 5-year retrospective of achievements. In Ajay Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*. Springer, 2008.

[19] Alan F. Smeaton, Paul Over, and Wessel 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.

[20] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. 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*, pages 421–430, New York, NY, USA, 2006. ACM Press.

[21] Cees G.M. Snoek, K.E.A. van de Sande, O. de Rooij, B. Huurnink, J.C. van Gemert, J.R.R. Uijlings, J. He, X. Li, I. Everts, V. Nedovic, M. van Liempt, R. van Balen, M. de Rijke, J.M. Geusebroek, Th. Gevers, M. Worring, A.W.M. Smeulders, D.C. Koelma, F. Yan, M.A. Tahir, K. Mikolajczyk, and J. Kittler. The mediamill trecvid 2008 semantic video search engine. In *Proceedings of the 8th TRECVID Workshop*, Gaithersburg, USA, November 2008.

[22] Hastie T. and Tibshirani R. Classification by pairwise coupling. Technical report, Standford University and University of Torronto, 1996.

[23] Rong Yan. *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval*. PhD thesis, Canegie Mellon University, 2006.

[24] Rong Yan and Alexander G. Hauptmann. The combination limit in multimedia retrieval. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 339–342, New York, NY, USA, 2003. ACM.

[25] Rong Yan and Alexander G. Hauptmann. A review of text and image retrieval approaches for broadcast news video. *Information Retrieval*, 10(4-5):445–484, October 2007.

[26] Jun Yang and Alexander G. Hauptmann. (un)reliability of video concept detection. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 85–94, New York, NY, USA, 2008. ACM.

[27] Wujie Zheng, Jianmin Li, Zhangzhang Si, Fuzong Lin, and Bo Zhang. Using high-level semantic features in video retrieval. In *Image and Video Retrieval*, volume Volume 4071/2006, pages 370–379. Springer Berlin / Heidelberg, 2006.