Combining Information Sources for Video Retrieval

The Lowlands Team at TRECVID 2003

Thijs Westerveld[◇] Tzvetanka Ianeva[†][*], Lioudmila Boldareva[‡], Arjen P. de Vries[◇] and Djoerd Hiemstra[‡]

[◇]Centrum voor Wiskunde en Informatica (CWI)
Amsterdam, The Netherlands
{thijs,arjen}@cwi.nl

[†]Departamento de Informàtica
Universidad de València
Valencia, Spain
tzveta.ianeva@uv.es

[‡]Department of Computer Science
University of Twente
Enschede, The Netherlands
{boldarli,hiemstra}@cs.utwente.nl

Abstract

The previous video track results demonstrated that it is far from trivial to take advantage of multiple modalities for the video retrieval search task. For almost any query, results on ASR transcripts have been better than any other run. This year's main success in our runs is that a combination of ASR and visual performs better than either alone! In addition we experimented with dynamic shot models, combining topic examples, feature extraction and interactive search.

1 Introduction

Often it is stated that a successful video retrieval system should take advantage of information from all available sources and modalities. Merging knowledge from for example speech, vision, and audio would yield better results than using only one of them. But previous video track results demonstrated that it is far from trivial to take advantage of multiple modalities for a video retrieval search task.

For this year's TRECVID workshop, we experimented with combining different types of information:

- combining different models/modalities
- combining multiple example images
- combining model similarity and human-judged similarity

The basic retrieval models we use to investigate the merging of different sources are the same models we used last year [9], they are described briefly in section 2 and more extensively in [10]. Section 2.1 describes a dynamic variant of the model that allows for describing spatio-temporal information as opposed to the spatial information captured in the basic models. These dynamic models can describe shots instead of still keyframes. For modelling the ASR transcripts,we use a basic language modelling approach (see Section 2.2). The interactive retrieval setup (Section 2.3) builds upon the visual models and language models. After the introduction of the separate models, we presents experiments for combining textual and visual information (Section 4), combining different examples (Section 5) and combining automatic similarity and interactive similarity (Section 7).

[*]The work has been carried out while at CWI, supported by grants GV FPIB01_362 and CTESRR/2003/43

2 Generative Probabilistic Retrieval Model

The retrieval model we use to rank video shots is a generative model inspired by the language modelling approach to information retrieval [6, 4] and a similar probabilistic approach to image retrieval [7]. We present – concisely – the visual part of the model, referring the interested reader to [10] for more details. The visual model ranks images by their probability of generating the samples (pixel blocks) in one or more query example(s) [10, 8]. The model is smoothed using background probabilities, calculated by marginalisation over the collection. So, a collection image ω_i is compared to an example image $\boldsymbol{x}$ consisting of N samples ($\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$) by computing its ability to explain the samples $\boldsymbol{x}$. The retrieval status value (RSV) of an image ω_i is defined as:

$$\mathrm{RSV}(\omega_i) = \frac{1}{N}\sum_{j=1}^{N} \log\left[\kappa P(x_j|\omega_i) + (1-\kappa)P(x_j)\right], \tag{1}$$

where κ is a mixing parameter.

Collection images ω_i are modelled as mixtures of Gaussians with a fixed number of components C [7, 10]:

$$P(\boldsymbol{x}|\omega_i) = \sum_{c=1}^{N_C} P(C_{i,c})\ \mathcal{G}(\boldsymbol{x}, \boldsymbol{\mu}_{i,c}, \boldsymbol{\Sigma}_{i,c}), \tag{2}$$

where N_C is the number of components in the mixture model, $C_{i,c}$ is component c of class model ω_i and $\mathcal{G}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian density with mean vector $\boldsymbol{\mu}$ and co-variance matrix $\boldsymbol{\Sigma}$:

$$\mathcal{G}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})},$$

where n is the dimensionality of the feature space and $(\boldsymbol{x}-\boldsymbol{\mu})^T$ is the matrix transpose of $(\boldsymbol{x}-\boldsymbol{\mu})$.

The samples are 8 by 8 pixel blocks, described by their DCT coefficients and their position in the image plane; the models are trained on these using standard EM [2]

2.1 Dynamic Model

The Dynamic model is a Gaussian Mixture Model in DCT-space-time domain. Instead of modelling just a single image (keyframe) we model a one-second video sequence around the keyframe as a single entity. The dynamic model is an extension of the static model, the sampling process is very similar. We take 29 frames around the keyframe and cut them in distinct blocks of 8 by 8 pixels. Each block is then described by its DCT coefficients, its x and y position in the image plane and its position in time (normalised between 0 and 1). Given this setup, the static model can be seen as a special case of the dynamic model where the temporal feature takes a fixed value of 0.5. The intuitive explanation in this case is: we do not know what is happening before and after the central time moment matching the keyframe.

The number of samples for training the dynamic model is much larger than for the static model (i.e. 29 times as large). The training process remains the same: feature vectors are fed to the EM algorithm to find the parameters μ_c, Σ_c, and $P(C_c)$. We use diagonal covariance matrices (Σ_c), this means the components are aligned to the axes, thus we cannot capture all temporal information, but we do capture the appearance and disappearance of objects. Figures 1 shows an example video sequence and a visualisation of the resulting model. It is easy to see how well the model fits the data, the *tree* in the top left corner is only visible at the beginning of the sequence. The corresponding component in the model also disappears at about $t = 0.5$ In other words, the GMM captures the disappearing of the tree. This effect is impossible to be captured from the static model where time is not taking into account.

2.2 ASR

The ASR approach used the same hierarchical language model as last year [9]. We model video as a sequence of scenes, each consisting of a sequence of shots. The generative model mixes the different levels of the hierarchy, with models for shots and scenes. Given a query with N_t terms $\boldsymbol{q} = (q_1, q_2, \ldots, q_{N_t})$, the RSV of a shot ω_i is defined as:

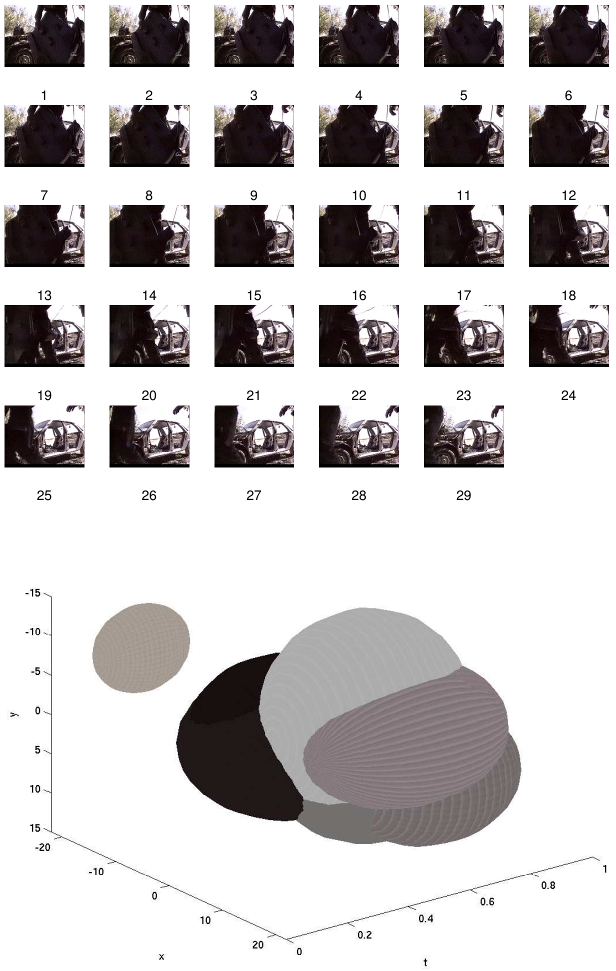

Figure 1: A shot represented by 29 frames around the keyframe (top) and a 3D visualisation of dynamic GMM computed from it (bottom). The bottom image shows mean colour and mean texture of the components where the standard deviation from the mean position in the $x-y-t$ is below 2. Prior probabilities and variance in colour and texture are not visualised.

$$\begin{aligned}\mathrm{RSV}(\omega_i) = \quad & \tfrac{1}{N_t}\textstyle\sum_{j=1}^{N_t}\log[\lambda_{\mathrm{Shot}}P(q_j|\mathrm{Shot}_i)+ \\ & \lambda_{\mathrm{Scene}}P(q_j|\mathrm{Scene}_i)+\lambda_{\mathrm{Coll}}P(q_j)] \\ & \text{with } \lambda_{\mathrm{Coll}} = 1-\lambda_{\mathrm{Shot}}-\lambda_{\mathrm{Scene}}\end{aligned} \tag{3}$$

Shot_i and Scene_i are the shot and scene to which ω_i belongs. The main idea behind this approach is that a good shot contains the query terms and is part of a scene having more occurrences of the query terms. Also, by including scenes in the ranking function, we hope to retrieve the shot of interest, even if the video's speech describes it just before it begins or just after it is finished. Because we only don't have scene boundaries, we assume pragmatically that each sequence of 5 consecutive shots forms a scene.

The features in the ASR model are simply the word tokens from the transcript. We estimate the foreground ($P(q_j|\mathrm{Shot}_i)$) and background ($P(q_j)$) probabilities in equation 3 by taking the term frequency and document frequency respectively [4]. We used the TREC-2002 video search collection to find the optimal values for the mixing parameters: $\lambda_{\mathrm{Shot}} = 0.090$, $\lambda_{\mathrm{Scene}} = 0.210$, and $\lambda_{\mathrm{Coll}} = 0.700$.

2.3 Interactive retrieval

In an interactive setting, the models above might be used as follows: First the user enters a text query that uses the speech recognition transcripts. As a result, he/she gets a screen full of video key frames. The user might now: *a*) rephrase his/her query, *b*) walk down the ranked list, i.e. look at the next screen of key frames, or *c*) use relevance feedback, i.e. retrieve key frames that are similar to one or more of the key frames currently on the screen. The text queries (see Section 2.2) can be evaluated almost instantly by the system. Whenever the system has to find similar images, however, the Gaussian mixture models of Equation 1 are used. It is well-known that similarity search using low level features does not scale very well. Given one example image, ranking the 32,000 key frames of the collection would take about 15 minutes on a fairly fast personal computer. Ranking 3.2 million key frames would take approximately 100 times as long.

To get reasonable on-line performance, we used a brute-force solution by precomputing for each key frame $\boldsymbol{x}$ its nearest neighbours ω_i (see e.g. [3]), according to the Gaussian mixture models. In practice, this means we have to calculate all image-image similarity pairs, and then take those image pairs that are nearest neighbours. The image pairs $(\boldsymbol{x}, \omega_i)$ are stored together with the probability $P(\boldsymbol{x}|\omega_i)$ defined by Equation 1 in an access structure called *association matrix*. Using the assocation matrix, we avoid expensive run time distance computations [1]. To search for images that are similar to two or more example images $\boldsymbol{x}_j$, the model's probabilities $P(\boldsymbol{x}|\omega_i)$ are combined by assuming conditional independence between selected images $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ given the hypothesised target image ω_i:

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots|\omega_i) \ = \ \prod_{j=1}^{n} P(\boldsymbol{x}_j|\omega_i) \qquad (4)$$

For images that are not among the nearest neighbours (that is, for which $P(\boldsymbol{x}_j|\omega_i)$ is not stored in the association matrix), we use a back-off mechanism, effectively replacing $P(\boldsymbol{x}_j|\omega_i)$ by a background model $P(\boldsymbol{x}_j)$. For the TREC experiments, the background model was assumed to be uniform.

Besides efficient on-line similarity computation, the association matrix serves another purpose: It can be trained from user interaction. By collecting user feedback, we hope to adapt the matrix in such a way that it represents user's associations between images. A possible interpretation of an image pair $(\boldsymbol{x}_j, \omega_i)$ in the matrix is the following: "Users that selected image ω_i also selected image $\boldsymbol{x}_j$"[1], or more precisely, 80 % of the users that selected image ω_i also selected image $\boldsymbol{x}_j$, if $P(\boldsymbol{x}_j|\omega_i) = 0.8$.

3 Experimental Setup

The search collection is indexed using the procedures described above. For each shot, we build a static model a dynamic model and a Language Model. Building queries from the topic descriptions is mostly automatic. The only manual action in constructing visual queries was selecting one or more image or video examples to be used for ranking. A textual query was constructed manually for each topic be taking only the content words from the topic description. From there on, the whole retrieval process was automatic, except for the experiments described in section 5.1. All image examples are rescaled to at most 272x352 pixels and then jpg compressed with a quality level of 20% to match size and quality of the collections videos. Further experiments are needed to test if this scaling and degrading of queries influences retrieval results. The interactive experiments only used textual queries; not the image examples from the topics.

4 Combining Modalities

If we (unrealistically) assume textual and visual information are independent, we can easily compute a joint probability of observing query text and visual example from a document. Under this assumption, we can simply multiply the individual probabilities (or sum the log probabilities from Equations 1 and 3). In previous TREC experiments, we found that such a combination strategy works well provided that the individual runs each have useful results [9]. This year, this seems to be the case, A combination of the dynamic models described in Section 2.1 and ASR (Section 2.2) outperforms the individual runs (see Table 1). The static run performs worse than ASR only, even though the MAP score for static only is the same as that for dynamic only. MAP scores hide a lot of information though. The dynamic run has a higher initial precision (See Figure 2), and thus in some ways, the dynamic run is better than the static run and therefore more useful in a combination. While the dynamic models represent some of the temporal aspects of a shot, like objects (dis-)appearing, the main advantage over the static models seems to result from two (related) aspects: more training data describing the visual content, and less dependency on choosing an appropriate keyframe.

Results of the dynamic+ASR are further improved by filtering out shots with anchor persons. These

[1] As in collaborative filtering used by Amazon.com: "Customers who bought this book also bought:"

Description	MAP
ASR only	.130
static only	.022
static + ASR	.105
dynamic only	.022
dynamic + ASR	.132
ARS + noAnchor	.133
static + noAnchor	.022
static + ASR + noAnchor	.106
dynamic + noAnchor	.022
dynamic + ASR + noAnchor	.135

Table 1: Mean Average Precision (MAP) for ASR only, visual only and combined runs (static and dynamic models).

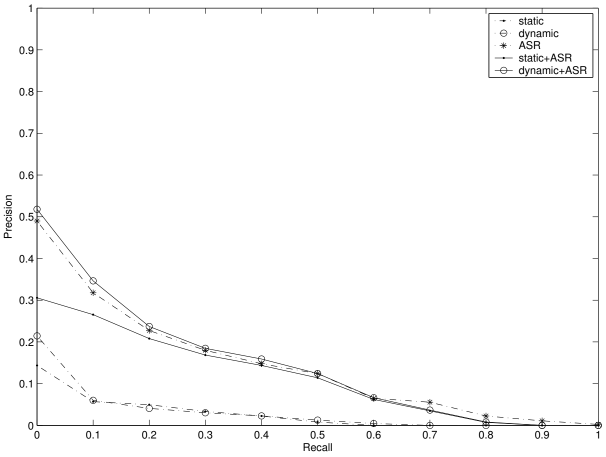

Figure 2: Recall-Precision graph for static, dynamic and ASR runs as well as multimodal combinations.

shots are identified by computing the likelihood of generating the set of frames annotated with the labels *news_person_face* and *studio_setting* from the shot models.[2] The 1077 shots[3] with the highest likelihood are assumed to contain anchor persons and are filtered out. The resulting MAPs are shown in Table 1

5 Combining Topic Examples

5.1 Merging Run Results

Last year we found combining multiple examples in one query is far from trivial [9]. Conflicting examples (like different views under different weather conditions) can easily cause bad results. Still, if we use just one 'best' topic example as a query there's a risk of missing a lot of relevant material (suppose for example we selected a close up shot of a point being scored in basketball, and most of the relevant shots in the collection happen to be overview shots of the playing field). This year instead of combining multiple examples in a single query, we experiment with running separate queries for each example and merging run results afterwards. For each topic we manually select a set of 'good' examples run separate queries for each example and then merge the results using a simple round-robin approach. Duplicates are filtered out afterwards.

In addition to this within topic merging of examples, we also experiment with merging results from different models. The goal here is to find out if it is possible to decide in advance what would be a good model to use for a given topic. When there are only still image examples, one could decide to use the static models, whilst the dynamic model might be used for video examples. For each topic, we manually selected a visual model to be used. The main strategy is the one described above: dynamic models for video examples and static ones (more precisely topic models, see Section 5.2) for still image examples. In addition, for topics where this seems apropriate, we filter out unwanted shots from the results set. The approach is similar to the one described in [5]. We use *Anchor Person*, *people*, *studio setting* and *weather maps* filters and remove those shots that have a high likelihood[4] of explaining annotated samples of

[2] For computational reasons, we computed the likelihood of a subset of the samples from the frames, in stead of the likelihood of the full set.

[3] After 1077 documents, we noticed a drop in Anchor person likelihood scores.

[4] High means within some top K, where K is set using sep-

the given feature (See also section 6).

The MAP of this merging run is .039, the highest among the runs that use only visual information, even though, for some topics, it contains results from the disappointing topic Model run (See Section 5.2). Further research is needed to find out how much of this success can be attributed to the combination of different examples and how much to the topic specific model selection.

5.2 Building Topic Models

Instead of explaining the query samples from the document models (and combining the results for different examples afterwards), we could go the other way and explain document samples from query models. In this setting, documents that have a high likelihood of being generated from the query model are assumed to be relevant.

We use all available examples for a given topic to build a GMM. The assumption here is that different components in the GMM can capture different aspects of the information need, thus contrasting examples (e.g. day and night shots of city views) are no longer a problem. During training of the model we allowed topic samples to be explained by a background model; this way very common, non distinguishing query samples do not contribute as much to the model parameters. After building the models we can use our normal ranking function (Equation 1) only now the roles of query and document are reversed; now we have a set of document samples $\boldsymbol{x}$ that need to be explained from the query model ω_i.

Smoothing with background probabilities as in Equation 1 is not wise in this case. When we use smoothing, common samples get a high score. This is useful in the standard query likelihood approach (Section 2), since these high background scores are independent of the document model under consideration; the contribution of the individual document models to scores of common samples, is therefore less important and the influence of these samples on the ranking is relatively small. However with the reversed likelihood that we are using here, smoothing means that common *document* samples get high scores. Obviously, this is *not* document independent and thus documents with a lot of common samples will get high scores and end up at the top of the ranked list. For this reason, we do not smooth the scores for the topic models and rank the documents using:

$$\mathrm{RSV}(\boldsymbol{x^D}) = \frac{1}{N}\sum_{j=1}^{N} \log P(x_j^D|\omega_Q), \tag{5}$$

where $\boldsymbol{x^D}$ is the set of N samples from document D and ω_Q is the model built from the samples from topic Q. Using this approach we obtain a MAP of only .005 and for many topics we retrieve a lot of black frames. Apparently, even without the smoothing, we retrieve mainly shots with common samples. This is not too surprising, since shots with common samples which are easily explained by *any* model, are often also well explained by a specific query model.

In an additional (not submitted) experiment, we calculate the odds rather than the likelihood of the document samples:

$$\begin{aligned} \mathrm{RSV}(\boldsymbol{x^D}) &= \frac{\sum_{j=1}^{N} \log P(x_j^D|\omega_Q)}{\sum_{j=1}^{N} \log P(x_j^D|\neg\omega_Q)} \\ &= \frac{\sum_{j=1}^{N} \log P(x_j^D|\omega_Q)}{\sum_{j=1}^{N} \log P(x_j^D)}. \end{aligned} \tag{6}$$

Using this approach, we get a MAP of .013, significantly higher than for the document likelihood ranking discussed above, but lower than the original query likelihood ranking. Perhaps we still have a background influence and are now confronted with a preference for uncommon documents or maybe the topic model can not capture the aspects of all the different examples in a single model with only 8 components. Further analysis is necessary.

6 Feature Extraction

In a way, feature extraction is the same as retrieval. There is a more or less coherent set of examples (annotated images) for a specific information need (feature to be detected). Thus, it is possible to use the techniques developed for search on the feature extraction task. The results could be regarded as a baseline

arate training data.

for systems that put more effort into building detectors for specific features.

Starting from the results of the collaborative annotation effort, we construct sets of examples for each specific feature, by grouping labels. For example everything annotated as either *building* or *house* is used in the example set for the *building* feature. The set of examples is then converted into feature vectors using the usual procedure (computing DCTs from 8x8 pixel blocks). From each set of samples, we take a random subset[5] and use that as the set of samples to use our search techniques on. We used the sample likelihood approach (See Section 2) and the reversed likelihood approach based on feature models both with and without using the background probabilities during training (Section 5.2). For some features some of the approaches got results above median, but overall the results are disappointing.[6] Possible reasons for this are the high diversity in the set of annotated examples for a given feature and again the influence of the background probabilities (cf. Section 5.2). Further experiments with odds rather then likelihoods and more carefully selected examples are needed to test whether feature extraction as search is in principle a reasonable option.

7 Interactive Experiments

The interactive experiments build on results from the automatic models as described in Section 2.3. Since computing the association matrix is computationally very intensive, we computed the likelihood using a random subset of 100 image samples.[7] Based on experiments on the TREC 2002 video collection on which we simulated user feedback using the assessments [1], we identified the following four experiments:

1. **Random** The user gets random key frames on the screen;

2. **Text-Only** The next screen of key frames is only based on the textual query, not on the user's feedback;

3. **Feedback** The next screen of key frames is based on the user's feedback using the association matrix based on the mixture models;

4. **Feedback-trained** The next screen of key frames is based on the user's feedback using an association matrix that was trained on the user feedback gathered by experiments 1 to 3.

The user interface of all four systems are exactly the same. The screen contains 12 images, three rows of four. Below each image, the user can check a box if he/she wants to have similar images. Only systems 3 and 4 actually use the feedback, the other systems record the feedback (for training system 4) but ignore it when updating the screen. Systems 2, 3 and 4 started out with exactly the same textual query, which was automatically taken from the topic descriptions without the subjects being aware of this, i.e., the systems started out with the same 12 images.

Four subjects, each did 15 minute search sessions for all topics. Three subjects each used the systems 1, 2 and 3 for one third of the topics in such a way that each topic was done on each system. They did not know which system was used on which topic. The fourth subject did all topics using system 4.

The results that were submitted to TREC were produced by taking the images that were selected by the user, and concatenate to that the ranking of the model at the last iteration step. Figure 3 contains the recall-precision graphs of systems 1–4. As expected, the text run performed much better than the random run on all recall levels (average precision 0.233 vs. 0.0557). On most recall levels, the feedback run was better than the text run (average precision 0.267 vs. 0.233). The feedback run on the trained matrix showed worse mean average precision performance than the feedback run on the visual feature matrix (average precision 0.240 vs. 0.267).

[5] We take at most 10,000 samples per feature, to keep everything tractable. For sets with fewer than 10,000 annotated samples, all available samples are used.

[6] Even a random system scores above median on some topics.

[7] A small experiment indicated that using a subset of 100 samples only, already gives a ranking similar to the one obtained by using the full set of samples.

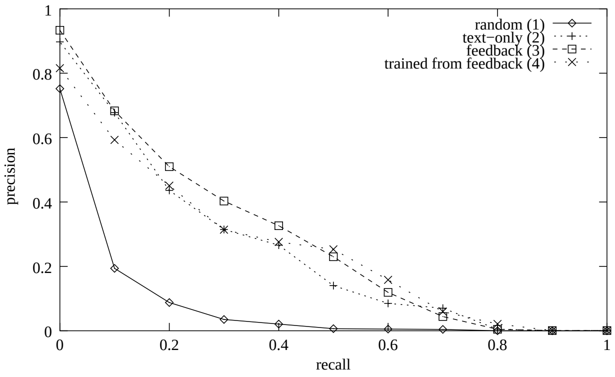

Figure 3: Precision at various recall levels of three systems: random, text-only and feedback

system	seen	pos.	assessed	agree-	pos	good
		feedb.	relevant	ment	seen	seen
random	3456	37	22	59.5 %	1.1 %	0.6 %
text-only	22908	266	227	85.3 %	1.2 %	1.0 %
feedback	17964	381	300	78.7 %	2.1 %	1.7 %
trained	14184	638	413	64.7 %	4.5 %	2.9 %

Table 2: Statistics of interactive runs

Table 2 contains for each system respectively the total number of images that have been displayed[8]; the number of images that were marked by the users as relevant to the search topic; and the number of those images that were assessed as relevant by the TREC assessors. Furthermore, it shows the percentage of agreement between our subjects and the TREC assessors; the percentage of seen images that were marked as relevant; and the percentage of seen images that are relevant according to TREC. Interestingly, the subjects identified 381 relevant images using the feedback system, whereas the subjects identified only 266 relevant images using the text-only system. This cannot be explained by differences between users: The users from a second group who did the exact same task (of which the results were not sent to TREC) showed similar results. From the statistics in Table 2 it appears that relevance feedback has positive effect, but for some reason the TREC evaluation measures show only a marginal difference. We hope to report a more detailed evaluation, measuring the performance on each feedback iteration step, in the final paper.

8 Conclusions

This paper presented the models used and experiments carried out for the TRECVID 2003 workshop. We focused on combining information on different levels. We combined information from different sources and modalities, information from different visual examples and human and model based information.

We extended our generative probabilistic models to include temporal information and found this improves the results. Combining these results with results from a ASR run gives another improvement. Whenever both text results (from ASR language models) and visual results (from GMM models) do something useful, a combination gives even better results.

Manually selecting good visual examples and useful models for a given topic gave the best results among the visual runs. The selection of (multiple) good examples and their combination are the main cause for this success.

Intuitively, building a model from all (or some) visual topic examples and ranking the documents by their probability of being generated from such a topic model, seems at least as good an approach as the reversed process (trying to explain query samples from document models). In practise however, this approach has some difficulties and documents with either many common or many uncommon samples (depending on the normalisation) seem to dominate the results. The same problems occur in the Feature Extraction task which we regard as a normal search task.

The interactive search results are obviously far better, than the manual ones. But also here we noticed that a combination of textual and visual information performed better than text alone.

[8] due to a bug, the random system was much slower than the other systems

References

[1] L. Boldareva, D. Hiemstra, and W. Jonker. Relevance feedback in probabilistic multimedia retrieval. In *DELOS Workshop on Multimedia Contents in Digital Libraries*, 2003. http://www.music.tuc.gr/MDL/.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977.

[3] R. Fagin. Fuzzy queries in multimedia database systems. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 1–10, 1998.

[4] D. Hiemstra. *Using language models for information retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

[5] T. I. Ianeva, A. P. de Vries, and H. Röhrig. Detecting cartoons: a case study in automatic video-genre classification. In *2003 IEEE International Conference on Multimeda & Expo*, 2003.

[6] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 275–281, 1998.

[7] N. Vasconcelos. *Bayesian Models for Visual Information Retrieval*. PhD thesis, Massachusetts Institut of Technology, 2000.

[8] T. Westerveld. Probabilistic multimedia retrieval. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR2002)*, pages 437–438, 2002.

[9] T. Westerveld, A. P. de Vries, and A. van Ballegooij. CWI at the TREC-2002 video track. In E. M. Voorhees and D. K. Harman, editors, *The Eleventh Text REtrieval Conference (TREC-2002)*, 2003.

[10] T. Westerveld, A. P. de Vries, A. van Ballegooij, F. M. G. de Jong, and D. Hiemstra. A probabilistic multimedia retrieval model and its evaluation. *EURASIP Journal on Applied Signal Processing*, 2003(2):186–198, 2003. special issue on Unstructured Information Management from Multimedia Data Sources.