

# Multimedia Retrieval using Multiple Examples

Thijs Westerveld and Arjen P. de Vries

CWI, INS1, PO Box 94079, 1090 GB Amsterdam, The Netherlands

**Abstract.** The paper presents a variant of our generative probabilistic multimedia retrieval model that is suitable for information needs expressed as multiple examples. Results have been evaluated on the TRECVID 2003 collection.

## 1 Introduction

A commonly used paradigm in image and video retrieval is that of querying by example (QBE). An example document (image or video) is presented to the search engine, and similar documents are requested. A slightly modified form of this paradigm is adopted in the TRECVID video retrieval benchmarking effort [1]. An information request is called a *topic*. It consists of a textual description of the multimedia need accompanied by one or more image and/or video examples. The goal is to return a ranked list of shots that meet the information need.

Combining multiple visual examples to return one set (or ranked list) of similar documents can be problematic. Consider for example the topic shown in Figure 1. Here the information need is for shots of points being scored in basketball. The need is clarified by 6 different examples, some of them close-ups of the ball going through the basket, others showing overview shots of the playing field. No document will be highly similar to *all* examples. Clearly, we are looking for some sort of *OR*-functionality here; a query result should be similar to any of the examples, but not necessarily to all.

A common approach to handling multiple queries is to run separate queries for each example and combine the results afterwards. In such an approach, the final score for a document is a function of either the *scores* or the *ranks* for the individual examples [2–4]. It is however far from trivial to choose a combination function that works well for a variety of queries.

The present work leaves this approach and capture all the different facets of a set of query examples in a single *topic model*. For retrieval, all documents in a collection are compared to this single topic model and ranked accordingly. The



Fig. 1. Topic 101: ‘Find shots of a basket being made’.

rest of the paper is organised as follows. Section 2 describes a generative probabilistic approach to information retrieval. Section 3 discusses how this approach can be applied to image and video retrieval. Section 4 shows experimental results and Section 5 summarises our main conclusions.

## 2 Generative Probabilistic Retrieval

Following Sparck Jones et al. [5], and Lafferty and Zhai [6], we introduce random variables  $D$  and  $Q$  to represent a document and a query and an event  $r$  to represent ‘relevant’, and try to answer the following “Basic Question”: *What is the probability that this document is relevant to this query?* This probability of relevance,  $P(r|D, Q)$ , can be estimated indirectly using Bayes’ rule:  $P(r|D, Q) = P(D, Q|r)P(r)/P(D, Q)$ . For ranking documents, we may avoid estimation of  $P(D, Q)$  using the odds of relevance:

$$\frac{P(r|D, Q)}{P(\bar{r}|D, Q)} = \frac{P(D, Q|r)P(r)}{P(D, Q|\bar{r})P(\bar{r})}, \quad (1)$$

where  $\bar{r}$  means not  $r$ . In the following,  $Q$  and  $D$  are assumed independent in the irrelevant case ( $\bar{r}$ ).

**Assumption 1.**  $P(Q, D|\bar{r}) = P(Q|\bar{r})P(D|\bar{r})$

Factoring the conditional probability  $P(D, Q|r)$  in different ways leads to two distinct, though probabilistically equivalent, models [6]. One model corresponds to *query generation*, and the other to *document generation*.

The query generation model results from factoring  $P(D, Q|r)$  as  $P(D, Q|r) = P(Q|D, r)P(D|r)$ , giving the following odds of relevance:

$$\frac{P(r|D, Q)}{P(\bar{r}|D, Q)} = \frac{P(D, Q|r)P(r)}{P(D, Q|\bar{r})P(\bar{r})} = P(Q|D, r) \cdot \underbrace{\frac{P(D|r)}{P(D|\bar{r})}}_{\text{prior odds}} \cdot \underbrace{\frac{P(r)}{P(Q|\bar{r})P(\bar{r})}}_{\text{independent of } D} \quad (2)$$

Since the goal is to rank documents, we can safely ignore the document independent terms. Also, we assume equal priors, i.e., a priori all documents are equally likely. This results in the following retrieval status value (RSV) for a document  $D$ :

$$\text{RSV}(D) = P(Q|D, r) \quad (3)$$

The document generation model results from factoring  $P(D, Q|r)$  as  $P(D, Q|r) = P(D|Q, r)P(Q|r)$ , arriving at a different equation for the odds of relevance:

$$\frac{P(r|D, Q)}{P(\bar{r}|D, Q)} = \frac{P(D, Q|r)P(r)}{P(D, Q|\bar{r})P(\bar{r})} = \frac{P(D|Q, r)}{P(D|\bar{r})} \cdot \underbrace{\frac{P(Q|r)P(r)}{P(Q|\bar{r})P(\bar{r})}}_{\text{independent of } D} \quad (4)$$

Ignoring all factors independent of  $D$  for ranking gives the following RSV:

$$\text{RSV}(D) = \frac{P(D|Q, r)}{P(D|\bar{r})} \quad (5)$$

### 3 Generative Multimedia Retrieval

The next step is to define how to estimate the probabilities  $P(Q|D, r)$ ,  $P(D|Q, r)$  and  $P(D|\bar{r})$ . Documents in our case are video shots and queries are either (sets of) images or shots. We choose to represent a shot by a representative keyframe, thus all queries and documents are images.<sup>1</sup> We estimate the (conditional) probabilities of queries and documents, by building a statistical model for each image.

#### 3.1 Gaussian Mixture Models

The model assumes that an image is the outcome of a random process that generates  $n$ -dimensional feature vectors  $\mathbf{x} = (x_1, \dots, x_n)$ , where each feature vector describes a small, square block of pixels. The retrieval framework itself is independent of the specificities of the features; we have used DCT coefficients and  $x$ - and  $y$ -coordinates to capture colour, texture and position of a pixel block. In the remainder the term *sample* is used to refer to both the feature vectors and the pixel blocks they describe. One or more images are represented as a bag of samples  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_S}\}$ .

The samples are assumed to be generated by a mixture of Gaussian sources, where the number of Gaussian components  $N_C$  is fixed for all images in the collection. The Gaussian mixture model (GMM) is fully described by a set of parameters  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{N_C})$  defining the different components. Each component  $C_i$  is described by its prior probability  $P(C_i)$ , the mean  $\boldsymbol{\mu}_i$  and the variance  $\boldsymbol{\Sigma}_i$ , thus  $\boldsymbol{\theta}_i = (P(C_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ . Details about estimating these parameters are deferred to Section 3.2. Equation 6 defines the probability of drawing one sample  $\mathbf{x}$  from a GMM with parameters  $\boldsymbol{\theta}$ .

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{i=1}^{N_C} P(C_i) \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_i|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \quad (6)$$

The probability of drawing a bag of samples is simply the joint probability of drawing the individual samples:

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N_S} p(\mathbf{x}_i|\boldsymbol{\theta}) \propto \sum_{i=1}^{N_S} \log p(\mathbf{x}_i|\boldsymbol{\theta}) \quad (7)$$

Working in the log domain avoids numerical underflow in the multiplication of many small numbers.

#### 3.2 Parameter Estimation

One way to look at mixture modelling for images is by assuming an image can show only so many different things, each of which is modelled by a Gaussian

---

<sup>1</sup> For a variant of the generative models in which we incorporate temporal aspects of video, we refer the reader to [3].

distribution. Each sample in a document is then assumed to be generated from one of these Gaussian components. This viewpoint, where ultimately each sample is explained by one and only one component, is useful when estimating the GMM parameters. The assignment of samples  $\mathbf{x}_j$  to components  $C_i$  can be viewed as hidden variables, so the Expectation Maximisation (EM) algorithm [7] can be used. This algorithm iterates between estimating the a posteriori class probabilities for each sample (the E-step) given the current model settings, and re-estimating the components parameters based on the sample distribution and the current sample assignments (the M-step):

**E-step:** Estimate the hidden assignments  $h_{ij}$  of samples  $x_j$  to components  $C_i$ , for all samples and components.

$$h_{ij} = P(C_i|\mathbf{x}_j) = \frac{p(\mathbf{x}_j|C_i)P(C_i)}{\sum_{c=1}^{N_C} p(\mathbf{x}_j|C_c)P(C_c)} \quad (8)$$

**M-step:** Update the component's parameters to maximise the joint distribution of component assignments and samples.  $\boldsymbol{\theta}^{\text{new}} = \arg \max_{\boldsymbol{\theta}} p(\mathcal{X}, \mathbf{H}|\boldsymbol{\theta})$ , where  $\mathbf{H}$  is the matrix with all sample assignments  $h_{ij}$ . More specifically:

$$\boldsymbol{\mu}_i^{\text{new}} = \frac{\sum_j h_{ij} \mathbf{x}_j}{\sum_j h_{ij}}, \quad (9)$$

$$\boldsymbol{\Sigma}_i^{\text{new}} = \frac{\sum_j h_{ij} (\mathbf{x}_j - \boldsymbol{\mu}_i^{\text{new}})(\mathbf{x}_j - \boldsymbol{\mu}_i^{\text{new}})^T}{\sum_j h_{ij}}, \quad (10)$$

$$P(C_i)^{\text{new}} = \frac{1}{N} \sum_j h_{ij} \quad (11)$$

The algorithm is guaranteed to converge.

### 3.3 Smoothing

Typicalities are more interesting than commonalities. Smoothing is a technique for explaining the common query terms, to reduce their influence on the ranking[8]. The estimates of the GMM are smoothed using interpolation with a general, background distribution – this technique is known as Jelinek-Mercer smoothing [9]. The smoothed version of the likelihood for a single sample  $\mathbf{x}$  becomes (cf. Equation 6):

$$p(\mathbf{x}|\boldsymbol{\theta}) = \kappa \left[ \sum_{i=1}^{N_C} P(C_i) \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_i|}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)} \right] + (1 - \kappa)p(\mathbf{x}), \quad (12)$$

where  $\kappa$  is a mixture parameter that can be estimated on training data with known relevant documents. A useful way of thinking about this smoothed variant is the following. A sample from a smoothed GMM comes either from one of the  $N_C$  components, or from a general background model. The background density  $p(\mathbf{x})$  is estimated by marginalisation over all document models in a reference collection  $\mathcal{D}$ :

$$p(\mathbf{x}) = \sum_{d \in \mathcal{D}} p(\mathbf{x}|\boldsymbol{\theta}_d)P(d) \quad (13)$$

The reference collection  $\mathcal{D}$  can be the current collection, a representative sample of that, or, another *comparable* collection.

### 3.4 GMMs and the Retrieval Framework

In the GMM approach, each document  $D$  has 2 representations: a set of samples  $\mathcal{X}_D$  and a Gaussian mixture model  $\theta_D$  (the same holds for queries  $Q$ ). To relate this to the conditional probabilities as introduced in Section 2, we estimate the probability of  $A$  conditioned on  $B$  and  $r$ ,  $P(A|B, r)$ , as the probability that the model of  $B$  ( $\theta_B$ ) generates the samples of  $A$  ( $\mathcal{X}_A$ ). Furthermore, to estimate the probability of  $A$  conditioned on the irrelevant event ( $P(A|\bar{r})$ ), we use the joint background density of the samples of  $A$  (cf. Equation 13). Thus, the retrieval status values for query generation (Eq. 3) and document generation (Eq. 5) are estimated as

$$\text{RSV}_{Q\text{gen}}(D) = P(Q|D, r) \equiv P(\mathcal{X}_Q|\theta_D) \quad (14)$$

$$\text{RSV}_{D\text{gen}}(D) = \frac{P(D|Q, r)}{P(D|\bar{r})} \equiv \frac{P(\mathcal{X}_D|\theta_Q)}{P(\mathcal{X}_D)} \quad (15)$$

## 4 Experiments

We evaluated the query and document generation variant of the generative probabilistic retrieval framework on the TRECVID 2003 search task [1]. For each document in the collection, and for each query example, we build GMMs as described in Section 3.2. Since we are interested in multiple-example queries, we regard samples from all available query images as a single set of query samples. We study two variants to represent the sets of query samples  $\mathcal{X}_Q$ . The first variant uses all available query samples, the second only those samples occurring in manually selected, *interesting* regions.<sup>2</sup>

### 4.1 Results

We have two model variants (query generation and document generation) and two ways of building query sample sets (full and regions). This amounts to four different system variants. Each of these variants is evaluated in isolation, as well as in combination with textual information. In these combined runs, we use a query generation approach for the textual part of the query: What is the probability that the textual query terms are generated by the document model (The models are built from speech transcripts, for details see [10]).<sup>3</sup> Table 4.1 shows results for different experimental settings (the last column is explained in

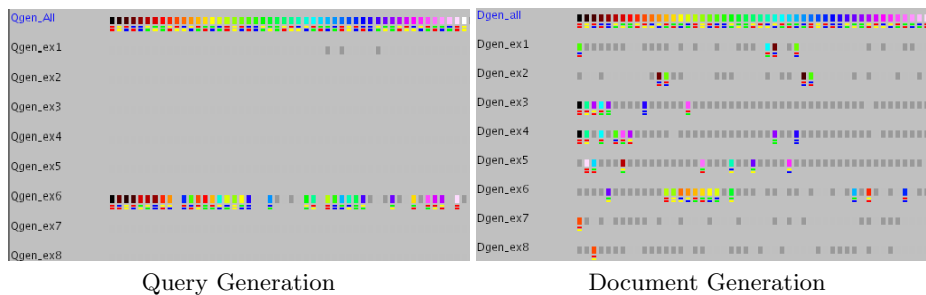
<sup>2</sup> Our manually selected query regions are available from <http://www.cwi.nl/projects/trecvid/>.

<sup>3</sup> A document generation approach for the textual part is problematic, since the short text queries we deal with have too little data to estimate proper topic models from.

Section 4.2). If we start with full example images, query generation outperforms document generation, but if we select regions, the situation is reversed. Looking at the average precision scores per topic, rather than only at the mean, and inspecting the returned ranked lists for the different models, it seems that we have to do with different types of matching. The query generation approach seems to be good at finding (near) exact matches, and is successful mainly when the set of examples is homogeneous (e.g. highly similar CNN baseball shots, or Dow Jones graphics). When a set of examples is less homogeneous, often a single example dominates the query generation results. Figure 2 shows this effect. In the document generation approach, the topic models seem to have learned important common aspects of the query examples, and thus all examples contribute to the combined result (see Figure 2), and more generic matches are found. The fact that common aspects are learned, could be an explanation why selecting regions helps here. When a user indicates important regions, the topic models will be more focused and retrieve better documents. In the query generation approach, selecting regions does not help, since exact matching relies heavily on background similarity.

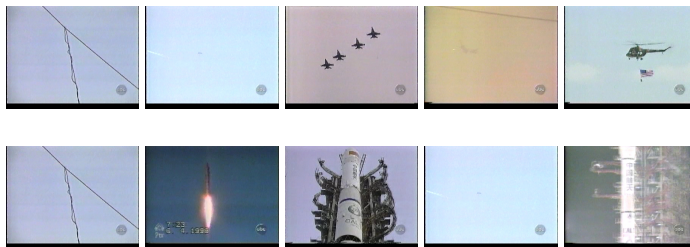
**Table 1.** Experimental results. MAP scores for using visual information only ( $MAP_{vis}$ ) and a combination of visual and textual information ( $MAP_{MM}$ ). The MAP for textual only is .130.

Qsamples	Qgen		Dgen		Dgen-BG	
	$MAP_{vis}$	$MAP_{MM}$	$MAP_{vis}$	$MAP_{MM}$	$MAP_{vis}$	$MAP_{MM}$
full	.028	.143	.026	.119	.034	.162
region	.023	.115	.026	.167	.034	.172



**Fig. 2.** Visualisations of the top N retrieved documents for the multiple-example run (top row) and the individual example runs, for the rocket launch query (topic0107). Created using NIST’s BeadPlot tool (see <http://www.itl.nist.gov/iaui/894.02/projects/beadplot/>). Colour codes represent document identifiers.

In combination with textual information, the region based document generation approach is better than any query generation variant. The lower performance of the query generation approaches can be explained because in this combination of modalities, the near exact matches on visual content interfere with the textual ranking. In the document generation approach however, the visual information seems to provide the generic visual context, while the textual information zooms in on specific results. For example, for topics that ask for airplanes, helicopters or rocket launches, the visual model captures the fact that we are looking for object against a background of sky. The textual information can then help distinguishing between the specific objects. Figure 3 shows an example.



**Fig. 3.** Document generation results (top 5) for Rocket launch query (topic107). The visual information sets the context (top rows, sky background) adding textual information fills in specifics (bottom, rockets)

## 4.2 Automatically Selecting Regions

It is clear that selecting regions is useful for the document generation approach. Rather than selecting them manually, it is possible to automatically select important parts of an example image. The main idea is to select those parts of the example that differ most from the average image. Samples that are likely to be generated by any model, should not influence the training process too much.

This can be achieved by incorporating background probabilities (Equation 13) in the training process. Again, hidden variables  $h_{ij}$  indicate the assignment of samples  $\mathbf{x}_j$  to components  $C_i$ , but now samples can also be assigned to the background, indicated by  $h_{BGj}$ . The EM-algorithm can be applied as before. The E-step changes to:

$$h_{ij} = P(C_i|\mathbf{x}_j) = \frac{p(\mathbf{x}_j|C_i)P(C_i)}{\sum_{c=1}^{N_C} p(\mathbf{x}_j|C_c)P(C_c) + p(\mathbf{x}_j)P(BG)} \quad (16)$$

$$h_{BGj} = P(BG|\mathbf{x}_j) = \frac{p(\mathbf{x}_j)P(BG)}{\sum_{c=1}^{N_C} p(\mathbf{x}_j|C_c)P(C_c) + p(\mathbf{x}_j)P(BG)}, \quad (17)$$

where  $P(BG|\mathbf{x}_j)$  is the posterior probability that  $\mathbf{x}_j$  is from the background, and  $P(BG)$  is the prior probability that we see background samples from the current model.

In the M-step, we do not update the background model  $p(\mathbf{x})$ . All we update are the component parameters (like in Equations 9,10, and 11), and the background prior ( $P(BG)$ ) for the current model.

$$P(BG)^{\text{new}} = \frac{1}{N} \sum_j h_{BGj} \quad (18)$$

Since common samples will be assigned to the background, only distinguishing samples are used in estimating the components' parameters. Figure, 4 shows an example image and the regions that are automatically selected to build the model from.



**Fig. 4.** Sphinx example. Original image (left) and samples selected by EM algorithm (right).

The rightmost column of Table 4.1 shows the results for the different settings trained using the new EM variant. A small (1%) sample from a comparable collection (the TRECVID 2003 development set) was used to estimate the background probabilities ( $P(\mathbf{x}_j)$ ) for the query samples. Clearly, using background probabilities during training helps. Automatically selecting regions using the new EM variant is almost as good as manually selecting important regions. Automatically finding distinguishing parts within manually selected regions, gives another improvement.

## 5 Conclusions

This work presented two ways of applying generative probabilistic retrieval models to the problem of video retrieval: a query generation approach and a document generation approach. We showed that the query generation approach is not good at handling multiple-example queries. Usually, there is no document model in the collection that is likely to generate all available visual examples. In such cases, the query generation approach results in a model that explains one of the examples very well.

The document generation approach on the other hand, has to capture all information available in the examples in a limited number of Gaussian components. Therefore, it will capture mainly things that are present in all examples, and thus build a model that describes the commonalities in the examples. This



leads to results that take all different examples into account. Often the things captured in the query models are of a generic, context-like nature (e.g. sky, grass, water). This turns out to be very useful in combination with textual information, where the results are far better than anything obtained so far using the query generation approach. We showed also, specifically for the document generation approach, that indicating important regions in example images is useful for retrieval. The automatic approach we developed yields comparable results to manual region selection. Automatically selecting important bits within manually created regions gives a slight improvement on the scores over using the user's manual selection as is.

Future work on the document generation model should prove whether indeed the results would be more like exact matches when multiple-example queries are modelled by more components. Another plan is to investigate automatic selection of samples for the query generation approach.

## References

1. Smeaton, A.F., Kraaij, W., Over, P.: Trecvid 2003 - an introduction. [11]
2. Jin, X., French, J.C.: Improving image retrieval effectiveness via multiple queries. In: Proceedings of the first ACM international workshop on Multimedia databases, ACM Press (2003) 86–93
3. Westerveld, T., Ianeva, T., Boldareva, L., de Vries, A.P., Hiemstra, D.: Combining information sources for video retrieval. [11]
4. Natsev, A., Smith, J.R.: Active selection for multi-example querying by content. In: IEEE International Conference on Multimedia and Expo (ICME). (2003)
5. Sparck Jones, K., Walker, W., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments – part 1. *Information Processing & Management* **36** (2000) 779–808
6. Lafferty, J., Zhai, C.: Probabilistic IR models based on document and query generation. In Croft, W.B., Lafferty, J., eds.: *Language Modeling for Information Retrieval*. Volume 13 of The Kluwer International Series on Information Retrieval. Kluwer Academic Publishers (2003)
7. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, series B* **39** (1977) 1–38
8. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (2001) 334–342
9. Jelinek, F., Mercer, R.: Interpolated estimation of markov source parameters from sparse data. In: Proceedings of the Workshop on Pattern Recognition in Practice. (1980)
10. Westerveld, T., de Vries, A.P., van Ballegooij, A., de Jong, F.M.G., Hiemstra, D.: A probabilistic multimedia retrieval model and its evaluation. *EURASIP Journal on Applied Signal Processing* **2003** (2003) 186–198 special issue on Unstructured Information Management from Multimedia Data Sources.
11. Smeaton, A.F., Kraaij, W., Over, P., eds. In Smeaton, A.F., Kraaij, W., Over, P., eds.: *TRECVID 2003 Workshop*, Gaithersburg, MD, USA, NIST, NIST Special Publications (2003)