Generative probabilistic models for multimedia retrieval: query generation versus document generation

Thijs Westerveld and Arjen P. de Vries
CWI/INS1, PO Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

This paper[1] presents the use of generative probabilistic models for multimedia retrieval. We estimate Gaussian mixture models to describe the visual content of images (or video) and explore different ways of using them for retrieval. We consider so-called *query generation* (how likely is the query given the document model) and *document generation* (how likely is the document given the query model) approaches and explain how both fit in a common probabilistic framework. We show that query generation is theoretically superior, and confirm this experimentally on the TRECVID search task. However, we found that in some cases a document generation approach gives better results. Especially in the cases where queries are narrow and visual results are combined with textual results, the document generation approach seems to be better at setting a visual context than the query generation variant.

[1]This is an extended and revised version of a previous conference paper [1].

1 Introduction

Many content-based multimedia retrieval tasks can be seen as decision theory problems. Clearly, this is the case for classification tasks, like face detection, face recognition, or indoor/outdoor classification. In all these cases a system has to decide whether an image (or video) belongs to one class or another (respectively face or no face; face A, B, or C; and indoor or outdoor). Even the ad hoc retrieval tasks, where the goal is to find *relevant* documents given a description of an information need, can be seen as a decision theory problem: documents can be classified into relevant and non-relevant classes, or we can treat each of the documents in the collection as a separate class, and classify a query as belonging to one of these. In all these settings, a probabilistic approach seems natural: an image is assigned to the class with the highest probability.[2]

In this paper, we take a *generative approach to information retrieval* —find the generating source of a piece of information. Such an approach has been applied successfully to retrieval problems involving various types of media, like language modelling for text retrieval [2, 3] and Gaussian mixture modelling for image retrieval [4, 5]. This paper compares and contrasts query generation and document generation approaches. In the query generation approach, the query is seen as an observation from one of the document models and we need to find the document model that most likely produced it. The document generation approach reverses this and estimates a model from the query. The goal is then to find the most likely documents given this model.

The remainder of this paper is organised as follows. Section 2 introduces the general probabilistic framework as it is used in amongst others text retrieval and derives two variants. Section 3 describes the generative image models and how they can be used in the different framework variants. Section 4 discusses the

[2] If some miss-classifications are more severe than others, a decision theoretic approach should be taken, and images should be assigned to the class with lowest risk.

theoretical differences between the variants and their expected behaviour. Section 5 reports on experimental results using the variants. Finally, Sections 6 and 7 discuss related work and the main conclusions of the present work respectively.

2 Probabilistic Retrieval Framework

Although Maron and Kuhns [6] were the first to consider probability theory for information retrieval, Robertson and Sparck Jones [7] were the first to put a probabilistic approach to use. To date, their binary independence retrieval model has been known as the classical probabilistic approach to information retrieval. The approach aims at directly estimating the odds of relevance given a query and a document representation. Sparck Jones et al. [8] present this classical probabilistic model starting from the following "basic question":

> What is the probability that *this* document is relevant to *this* query?

Lafferty and Zhai [9] start from the same basic question to show that this classical model is probabilistically equivalent to the modern language models for information retrieval [2, 3]. This section follows Lafferty and Zhai to show how these two probabilistic models relate to each other.

We start by introducing random variables D and Q to represent a document and a query, and a random variable R to indicate relevance. R can take two values: relevant $R = r$ or not relevant $R = \bar{r}$. In a probabilistic framework the basic question translates to estimating the probability of relevance $\mathrm{P}(r|D,Q)$.[3] This can be estimated indirectly using Bayes' rule:

$$\mathrm{P}(r|D,Q) = \frac{\mathrm{P}(D,Q|r)\mathrm{P}(r)}{\mathrm{P}(D,Q)} \tag{1}$$

[3]Random variables are omitted when instantiated, unless this may cause confusion. Thus $\mathrm{P}(r|D,Q)$ means $\mathrm{P}(R = r|D,Q)$.

For ranking documents, to avoid the estimation of $\mathrm{P}(D, Q)$, we may also estimate the odds:

$$\frac{\mathrm{P}(r|D,Q)}{\mathrm{P}(\bar{r}|D,Q)} = \frac{\mathrm{P}(D,Q|r)\mathrm{P}(r)}{\mathrm{P}(D,Q|\bar{r})\mathrm{P}(\bar{r})}. \tag{2}$$

As Lafferty and Zhai [9] show, two probabilistically equivalent models are obtained by factoring the conditional probability $\mathrm{P}(D, Q|r)$ in different ways. One model is based on query generation, the other on document generation.

2.1 Query generation

If $\mathrm{P}(D, Q|r)$ is factored as $\mathrm{P}(D, Q|r) = \mathrm{P}(Q|D, r)\mathrm{P}(D|r)$ we arrive at the following odds:

$$\frac{\mathrm{P}(r|D,Q)}{\mathrm{P}(\bar{r}|D,Q)} = \frac{\mathrm{P}(D,Q|r)\mathrm{P}(r)}{\mathrm{P}(D,Q|\bar{r})\mathrm{P}(\bar{r})} = \frac{\mathrm{P}(Q|D,r)\mathrm{P}(D|r)\mathrm{P}(r)}{\mathrm{P}(Q|D,\bar{r})\mathrm{P}(D|\bar{r})\mathrm{P}(\bar{r})} = \frac{\mathrm{P}(Q|D,r)\mathrm{P}(r|D)}{\mathrm{P}(Q|D,\bar{r})\mathrm{P}(\bar{r}|D)} \tag{3}$$

Under the assumption that Q and D are independent in the unrelevant case:

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

$\mathrm{P}(Q|D, \bar{r})$ reduces to $\mathrm{P}(Q|\bar{r})$. Keeping in mind that the goal is to rank documents for a single fixed query, allows us to ignore all factors that are independent of D. Thus, we arrive at the following retrieval status value (RSV):

$$\mathrm{RSV}_{\mathrm{Qgen}}(d) = \mathrm{P}(q|d,r)\frac{\mathrm{P}(r|d)}{\mathrm{P}(\bar{r}|d)} \tag{4}$$

Here, the first factor is query dependent, the second factor is the prior odds of a document being relevant. The prior odds could be based on surface features of the documents like format, source, or length. For example, photographic images may be more likely to be relevant than graphic images, CNN videos may be preferred over NBC ones, or long shots may have a higher probability of relevance than short ones. Surface features like these have proved successful in text retrieval and especially web search [10]. However, if no prior knowledge is available, a sensible option is to assume equal priors: a priori all documents are

equally likely. This reduces the RSV to

$$\text{RSV}_{\text{Qgen}}(d) = \text{P}(q|d,r) \tag{5}$$

This query generation variant is used in the language modelling approach to text retrieval [2, 3].

2.2 Document generation

Factoring $\text{P}(D,Q|r)$ differently, using $\text{P}(D,Q|r) = \text{P}(D|Q,r)\text{P}(Q|r)$, gives different odds:

$$\frac{\text{P}(r|D,Q)}{\text{P}(\bar{r}|D,Q)} = \frac{\text{P}(D,Q|r)\text{P}(r)}{\text{P}(D,Q|\bar{r})\text{P}(\bar{r})} = \frac{\text{P}(D|Q,r)\text{P}(Q|r)\text{P}(r)}{\text{P}(D|Q,\bar{r})\text{P}(Q|\bar{r})\text{P}(\bar{r})} \tag{6}$$

Under Assumption 1, and ignoring all factors independent of D, we arrive at the following RSV:

$$\text{RSV}_{\text{Dgen}}(d) = \frac{\text{P}(d|q,r)}{\text{P}(d|\bar{r})} \tag{7}$$

This document generation variant is the one used in the binary independence retrieval model [7, 8], although the dependence on Q is implicit there. In the binary independence retrieval model, probabilities are estimated based on term distributions in relevant and irrelevant documents.

3 Generative image models

The next step is to define how to estimate the probabilities $\text{P}(Q|D,r)$, $\text{P}(D|Q,r)$ and $\text{P}(D|\bar{r})$. Generative probabilistic models will be used to estimate these conditional probabilities: we build a statistical model for each document in the collection as well as for the queries. In the query generation approach, we then compute the probability of observing the query image from each of the document models, and use that for ranking. Figure 1 visualises this: from each document a model is built, visualised by showing the location, colour and texture of the

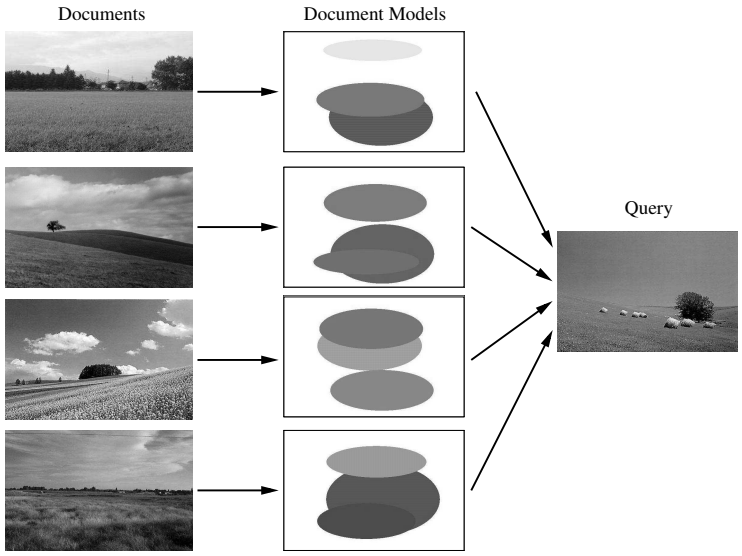

Figure 1: Visualisation of query generation framework.

model's components, and for each model the likelihood of generating the query is computed and used to rank the documents.

The document generation variant essentially reverses the process: a model is built from the query image and the likelihood of each document image given this query model is computed (see Figure 2).

The end of this section fills in the details of using generative image models in the probabilistic framework of the previous section. First, we introduce the generative image models and how to estimate them from data.

3.1 Gaussian mixture models

Documents in our case are video shots and queries are either images or shots. In this work, a shot is represented by a keyframe. A variant in which temporal aspects are incorporated is presented in [11]. We assume, each document (image) is composed of a set of small square blocks of pixels, each of them represented by a feature vector. Thus, an image is represented as a bag of samples $\mathcal{X} =$

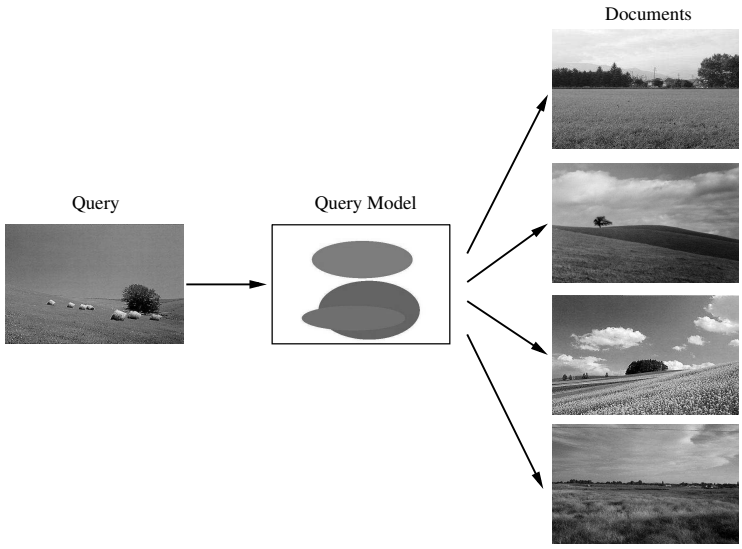

Figure 2: Visualisation of document generation framework.

$\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_{N_S}}\}$.[4] The generative models are independent of the nature of the feature vectors; we have used DCT coefficients and x- and y-coordinates to capture colour, texture and position of a pixel block.

We build a separate mixture model for each image in the collection. The idea is that the model captures the main characteristics of the image. The samples in an image are assumed to be generated by a mixture of Gaussian sources, where the number of Gaussian components C is fixed for all images in the collection. A Gaussian mixture model is described by a set of parameters $\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_C})$ each defining a single component. Each component c_i is described by its prior probability $\mathrm{P}(c_i|\boldsymbol{\theta})$, the mean $\boldsymbol{\mu_i}$ and the variance $\boldsymbol{\Sigma_i}$, thus $\boldsymbol{\theta_i} = (\mathrm{P}(c_i|\boldsymbol{\theta}), \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$. Details about estimating these parameters are described in the next subsection. The process of generating an image is assumed to be the following (see Figure 3):

1. Take the Gaussian mixture model $\boldsymbol{\theta}$ for the image

[4] In the following the term *sample* refers to both a pixel block, and the feature vector describing it.

2. For each sample $\boldsymbol{x}$ in the document

   (a) Pick a random component c_i from Gaussian mixture model θ according to the prior distribution over components $\mathrm{P}(c)$

   (b) Draw a random sample from c_i according to the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

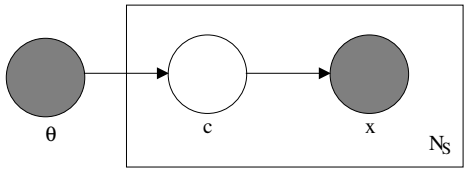

Figure 3: Graphical representation of Gaussian mixture model. Observerd variables are represented as solid nodes, hidden variables as open nodes. Arcs indicate dependencies ant the box stands for the repeated sampling of variables.

Here, $\boldsymbol{\theta}$ is an observed variable; the mixture model, from which the samples for a given image are drawn, is known. For a given sample however, it is unknown which component generated it, thus components are unobserved variables. The probability of drawing a sample $\boldsymbol{x}$ from a Gaussian mixture model with parameters $\boldsymbol{\theta}$ is thus defined as follows.

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{i=1}^{C} \mathrm{P}(c_i|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{x}|c_i, \boldsymbol{\theta}) \tag{8}$$

$$= \sum_{i=1}^{C} \mathrm{P}(c_i|\boldsymbol{\theta}) \; \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})} \tag{9}$$

The probability of drawing a bag of samples $\mathcal{X}$ is simply the joint probability of drawing the individual samples:

$$\mathrm{p}(\mathcal{X}|\boldsymbol{\theta}) = \prod_{\boldsymbol{x}\in\mathcal{X}} \mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta}) \tag{10}$$

3.2 Maximum likelihood estimates

To train a Gaussian mixture model from a given set of samples, i.e., to build a model for a document or query, a natural approach is to use maximum likelihood

estimates. Thus, the optimal model for a given document is that model that best explains the document's samples. For Gaussian mixture models it is hard to find this optimum analytically, but Expectation Maximisation [12] can be used as described below.

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 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 Gaussian mixture model parameters. The assignments of samples $\boldsymbol{x_j}$ to components C_i can be viewed as hidden variables, so the Expectation Maximisation (EM) algorithm can be used. This algorithm iterates between estimating the a posteriori class probabilities for each sample given the current model settings (the E-step), 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.

$$h_{ij} = \mathrm{P}(C_i|\boldsymbol{x_j}) = \frac{\mathrm{p}(\boldsymbol{x_j}|C_i)\mathrm{P}(C_i)}{\sum_{c=1}^{N_C} \mathrm{p}(\boldsymbol{x_j}|C_c)\mathrm{P}(C_c)} \tag{11}$$

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

$$\boldsymbol{\mu_i}^{\mathrm{new}} = \frac{\sum_j h_{ij}\boldsymbol{x_j}}{\sum_j h_{ij}}, \tag{12}$$

$$\boldsymbol{\Sigma_i}^{\mathrm{new}} = \frac{\sum_j h_{ij}(\boldsymbol{x_j}-\boldsymbol{\mu_i}^{\mathrm{new}})(\boldsymbol{x_j}-\boldsymbol{\mu_i}^{\mathrm{new}})^T}{\sum_j h_{ij}}, \tag{13}$$

$$\mathrm{P}(C_i)^{\mathrm{new}} = \frac{1}{N}\sum_j h_{ij} \tag{14}$$

The algorithm is guaranteed to converge to a local optimum. Previous experiments suggest EM initialisation hardly influences the retrieval results [13], but

more research is needed to verify this.

3.3 Smoothing

When the models are estimated on little data, there is the risk that the estimates are not accurate. Especially, in generative models on discrete data, like the language models used in text retrieval [2, 3], there is the *zero-frequency problem*: unseen events get zero probability. Therefore, language modelling approaches usually apply some sort of smoothing on the estimates.

The zero-frequency problem does not exist with Gaussian mixture models, since Gaussians have infinite support, but smoothing also serves another purpose, namely that of explaining common query terms and reducing their influence on the ranking [14]. This second function of smoothing is also useful in image retrieval: general query samples should not influence the ranking too much (typicalities are more interesting than commonalities). To smooth the estimates for the Gaussian mixture model, interpolation with a general, background distribution is used. This is also called Jelinek-Mercer smoothing [15]. The smoothed version of the likelihood for a single sample $\boldsymbol{x}$ becomes:

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta}) = \kappa \left[\sum_{i=1}^{N_C} \mathrm{P}(C_i) \ \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1} (\boldsymbol{x}-\boldsymbol{\mu_i})} \right] + (1-\kappa)\mathrm{p}(\boldsymbol{x}), \quad (15)$$

where κ is a mixture parameter that can be estimated on training data with known relevant documents. The background density $\mathrm{p}(\boldsymbol{x})$ is estimated by marginalisation over all document models in a reference collection $\mathcal{D}$:

$$\mathrm{p}(\boldsymbol{x}) = \sum_{d \in \mathcal{D}} \mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta_d}) \mathrm{P}(d) \quad (16)$$

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

3.4 Generative image models and the retrieval framework

In the Gaussian mixture modelling approach, each document d has 2 representations: a set of samples $\mathcal{X}_d$ and a Gaussian mixture model $\boldsymbol{\theta}_d$ (the same holds for queries Q). To relate this to the conditional probabilities introduced in Section 2, we estimate $\mathrm{P}(A|B,r)$ as the probability that the model of B ($\boldsymbol{\theta}_B$) generates the samples of A ($\mathcal{X}_A$). Furthermore, to estimate $\mathrm{P}(A|\bar{r})$ we use the joint background density of all samples of $\mathcal{X}_A$ (cf. Equation 16). Thus, the retrieval status values for query generation (5) and document generation (7) are estimated as

$$\mathrm{RSV}_{\mathrm{Qgen}}(d) = \mathrm{P}(q|d,r) \equiv \mathrm{P}(\mathcal{X}_q|\boldsymbol{\theta}_d) \tag{17}$$

$$\mathrm{RSV}_{\mathrm{Dgen}}(d) = \frac{\mathrm{P}(d|q,r)}{\mathrm{P}(d|\bar{r})} \equiv \frac{\mathrm{P}(\mathcal{X}_d|\boldsymbol{\theta}_q)}{\mathrm{P}(\mathcal{X}_d)} \tag{18}$$

4 Query generation versus document generation

Theoretically, using document generation for ranking is not ideal. Intuitively, a document that has exactly the same distribution as the query model should get the highest retrieval status value. However, as the following analysis of the RSV function shows, in the document generation approach, other documents are favoured.

$$\mathrm{RSV}_{\mathrm{Dgen}}(d) = \prod_{\boldsymbol{x}\in\mathcal{X}_d}\left[\frac{\kappa\mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta}_q)}{\mathrm{p}(\boldsymbol{x})} + (1-\kappa)\right] \leq \prod_{\boldsymbol{x}\in\mathcal{X}_d}\max_{x'}\left[\frac{\kappa\mathrm{p}(\boldsymbol{x'}|\boldsymbol{\theta}_q)}{\mathrm{p}(\boldsymbol{x'})} + (1-\kappa)\right] \tag{19}$$

Thus, the (hypothetical) document that is a repetition of the single most likely sample will receive the highest RSV. In practise, this means that the query model component with the largest prior will dominate the results. For example, if a query consists of 60% grass and 40% sky, the document generation approach will prefer documents that show only grass.

The query generation approach does not suffer from this problem, since it searches for the most likely model instead of the most likely set of samples. The

fact that an observation consisting of a repetition of a single sample gets the highest likelihood for a given document model is irrelevant, since we are looking at a single fixed observation (the set of query samples). To get a high score, a document model should explain all these samples reasonably well.

However, also in the query generation approach, a document with exactly the same distribution as the query will not receive the highest score, because of the smoothing. The RSV is computed based on the interpolation of foreground and background probabilities. The model that maximises that distribution is not necessarily the same as the query model (which maximises foreground only). Intuitively, this means the model that gets the highest score in the query generation approach is the model that best explains the most distinguishing query samples. This may not be ideal, but it seems a more reasonable approach than document generation. The experiments described in the next section investigate whether indeed query generation outperforms document generation.

5 Experiments

The TRECVID2003 test collection [16] is used to compare the document and query generation variants. TRECVID is a workshop series with the goal of promoting progress in content-based retrieval from digital video via open, metrics-based evaluation. This paper focuses on TRECVID's search task, defined as follows:

> Given the search test collection, a multimedia statement of information need (topic), and the common shot boundary reference for the search test collection, return a ranked list of at most 1000 common reference shots from the test collection, which best satisfy the need.

The TRECVID2003 test collection consists of 65 hours of ABC and CNN news broadcasts from 1998. The collection is shot segmented and comes with

a predefined set of keyframes which we use to represent the shots. The 25 topics in the test collection are multimedia descriptions of an information need, consisting of a textual description and one or more image or video examples. For each topic, relevance judgements are available; these indicate which shots are relevant for the topic.

5.1 Experimental setup

For each document in the collection, we use the set of document samples $\mathcal{X}_d$ to build a document model $\boldsymbol{\theta}_d$ as described in Section 3. These document models are used in the query generation approach. The same set of samples $\mathcal{X}_d$ is used in the document generation approach. The set of query samples $\mathcal{X}_Q$ is varied. We experiment with using *all* available examples or a manually selected *designated* example for each topic. In addition, we use either the full example images or only a manually selected *interesting* region.[5] Thus, in total four different sets of query samples for each topic exist (allEx-full, allEx-region, desEx-full and desEx-region).

For the document generation variant, topic models are built from the different sets of query samples (all/regions). The score for each document is computed as the likelihood of the set of (all) document samples using (18) and (15). The background probabilities are estimated over a small (1%) random sample from the comparable development set, available with the TRECVID2003 collection, and κ is set to 0.90, based on earlier experiments with the TRECVID2002 collection [4, 17].

In the query generation variant, a document model is built for each document in the collection. For each of the four variants of constructing a set of query samples, documents are ranked using their likelihood of generating that set of query samples (17).

[5] Designated examples and selected regions are available from `http://www.cwi.nl/projects/trecvid/`.

Table 1: Mean average precision scores for different system variants. Both the scores for using visual information only and the scores for a combination of visual and textual information are listed (mean average precision for textual only is .130).

	visual		visual+textual	
Qsamples	Qgen	Dgen	Qgen	Dgen
allEx-full	.028	.026	.143	.119
allEx-region	.026	.026	.142	.167
desEx-full	.025	.015	.134	.130
desEx-region	.022	.013	.134	.123

5.2 Results

We looked at results in a visual only situation, as well as in combination with results from a textual query. The textual description of a document (shot) comes from speech transcripts that have been made available by LIMSI [18]. To model the textual information, we follow a query generation approach.[6] To combine visual and textual information, we treat them independently and compute the joint probability of textual terms and visual samples (see [4, 17] for details on the textual models and the combination). Table 1 shows the results for the different settings.

The results show that in a mono-modal setting, query generation gives better results than document generation. This was to be expected given the comparison of the two approaches in Section 4. The effects are in particular evident in the designated example variants (desEx-full and desEx-region). Indeed, it is the case that results are dominated by the component with the highest prior. Figure 4 shows an example. The model captures both the dark blue background and the

[6] A document generation approach for the textual part is problematic, since the short text queries provide insufficient data to estimate proper topic models from.

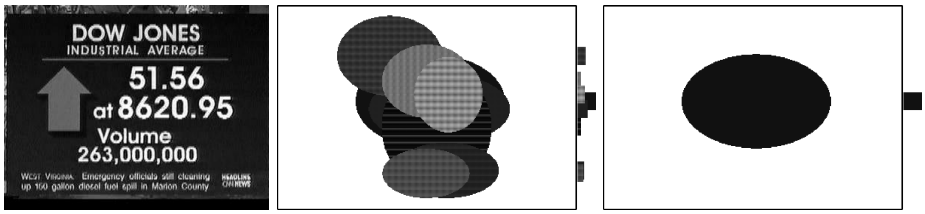

Figure 4: Designated example for VT0120: *Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day. The number of points risen that day must be visible*; a visualisation of the model built from that example; and the dominating component in the model (component with highest prior).

light textures in front of it, but since the document generation approach favours documents that match the most likely component, the top returned documents are mainly dark blue[7].

In the variants that use all available topic examples (allEx-full, allEx-region) the same effects play a role, but they do not disturb the results as much. Again, results may be dominated by a single component, but this component is likely to be more useful for satisfying the information need. Since the topic model in this case is built from multiple examples, the component with the highest prior is likely to capture (some of) the common aspects rather than an artifact of an individual example. Figure 5 shows the model built from all examples for the Dow Jones topic along with the component with the highest prior. The different examples for this topic (not shown here) vary in background, but all have light text and graphics in front. Hence, the dominating component is the one that captures the light textured foreground.

Although combining multiple examples helps in the document generation variant, the query generation variant gives better results still on all visual only tasks. However, in combination with textual information, document generation outperforms query generation when the query models are built from manually

[7]Verified by manually inspecting the results.

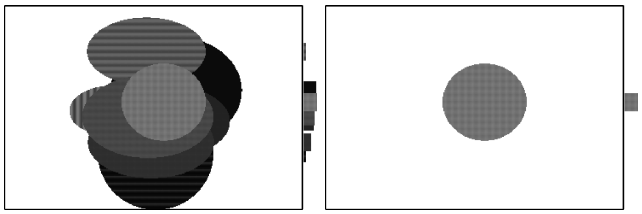

Figure 5: A visualisation of the model built from all VT0120 (*Dow Jones*) examples; and the dominating component in the model (component with highest prior).

selected regions. Further research is needed to understand this fully, but the following elements may play a role. Because regions are selected manually, the query model is relatively narrow, i.e., it describes a relatively homogeneous area.[8] Therefore, perhaps favouring documents containing repetitions of a few likely samples, as the document generation approach does, may be advantageous. Another possible explanation comes from the combination with the textual information. The visual content may set a context that can help to improve textual results. Highly ranked documents based on the visual document generation approach may show only the most dominating aspect of the query model (e.g. *sky*), but the textual information can help to re-rank the results, or to zoom in on relevant documents (e.g., *rockets*). See Figure 6 for an example.

6 Related Work

Several research groups have proposed to use Gaussian mixture densities to model visual information [19, 20, 21]. Both Vasconcelos and Lippman [19] and Greenspan et al. [20] model each of the images in a collection using a mixture of Gaussians. A query image is modelled like a document image, and the images are

[8] In earlier work we showed that automatically selecting *distinguishing* regions has a similar effect [1].

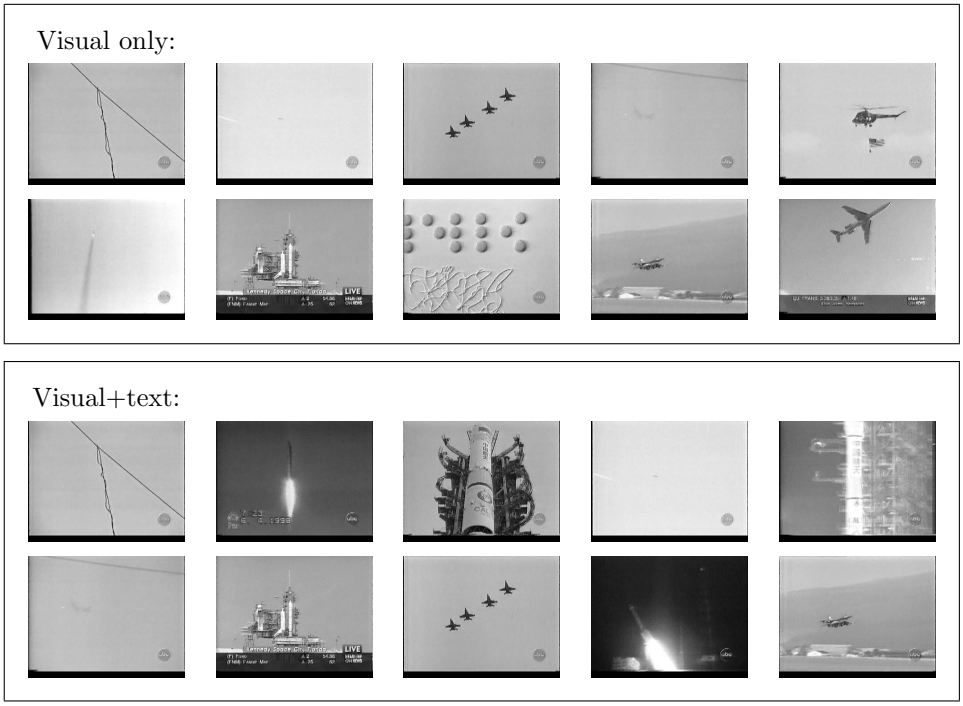

Figure 6: Document generation results for Rocket launch query (topic107). The visual information sets the context (top rows, sky background) adding textual information fills in specifics (bottom, rockets)

ranked using a measure of similarity between the query and document models. Vasconcelos and Lippman [19] approximate the likelihood that a random sample from the query model is generated from the document model. In later work, they develop approximations to the KL-divergence between query and document model, and use that for ranking [5]. Greenspan et al. extend their image model to one for video retrieval by incorporating a temporal dimension in their feature space [22, 23]. Luo et al. [21] also work with video material. They use Gaussian mixture densities to model predefined classes of medical video clips. For example, separate mixture models are estimated for surgery and diagnosis videos. Luo et al. use maximum likelihood classification to label unseen videos.

All these approaches either compare query and document models directly,

for example by using measures based on the KL-divergence, or they compute the likelihood of the query given document models. The latter is basically the query generation approach discussed in the present work. The document generation approach has to our knowledge not been applied to image retrieval before. For text retrieval, Lavrenko has experimented with document generation variants [24, Chapter 3], but with limited success.

Generative approaches have also been used to automatically annotate images [25, 26].

7 Conclusions

This paper presented two ways of applying generative probabilistic models to multimedia retrieval: a query generation approach and a document generation approach. We discussed the theoretical differences between the two approaches and argued query generation is closer to the intuitive behaviour of retrieving documents with a distribution of features similar to that in the query. Experimental results confirmed that indeed the top retrieved documents in the document generation approach often have a distribution that is quite different from the query, in fact the query is only partially matched. Remarkably, in some situations this behaviour gives better results in terms of mean average precision. This is the case when multiple examples are combined in a query, and when interesting regions within the query examples are selected manually. In such a situation, the partial match between the top documents and the query seems to be based on that part of the query that captures the information need best, i.e., the part that all examples have in common. Especially in combination with textual results, where the textual information can re-rank results and zoom in on relevant aspects, this document generation variant seems valuable, and results outperform all query generation combinations.

Finally, to favour documents that have a similar distribution as the query,

perhaps directly comparing query and document models using cross-entropy, or the Kullback-Leibler (KL) divergence, is a better approach than computing the likelihood that a document model generates the query samples or vice versa. However, KL is not analytically solvable for Gaussian mixture models. Approximations have been proposed [5], but in generic collections the underlying assumptions are violated and results may be sub-optimal [4, 17]. More research is needed to find alternative ways of comparing distributions based on Gaussian mixture models.

References

[1] Thijs Westerveld and Arjen P. de Vries. Multimedia retrieval using multiple examples. In *Proceedings of The International Conference on Image and Video Retrieval (CIVR2004)*, Dublin, Ireland, 2004.

[2] J. M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281. ACM Press, 1998. ISBN 1-58113-015-5.

[3] Djoerd Hiemstra. A linguistically motivated probabilistic model of information retrieval. In Christos Nicolaou and Constantine Stephanidis, editors, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, volume 513 of *Lecture Notes in Computer Science*, pages 569–584. Springer-Verlag, 1998.

[4] Thijs Westerveld. *Using generative probabilistic models for multimedia retrieval*. PhD thesis, University of Twente, November 2004. URL `http://purl.org/utwente/41716`.

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

[6] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–244, 1960. ISSN 0004-5411.

[7] Stephen Robertson and Karen Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[8] K. Sparck Jones, W. Walker, and S. Robertson. A probabilistic model of information retrieval: development and comparative experiments, parts 1 & 2. *Information Processing & Management*, 36:779–840, 2000.

[9] John Lafferty and Chengxiang Zhai. Probabilistic IR models based on document and query generation. In W. Bruce Croft and John Lafferty, editors, *Language Modeling for Information Retrieval*, volume 13 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers, 2003.

[10] Wessel Kraaij, Thijs Westerveld, and Djoerd Hiemstra. The importance of prior probabilities for entry page search. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Järvelin, editors, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34, Tampere, Finland, 2002. ACM Press. ISBN 1-58113-561-0.

[11] Tzvetanka Ianeva, Arjen P. de Vries, and Thijs Westerveld. A dynamic probabilistic retrieval model. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2004. to appear.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from

incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977.

[13] Thijs Westerveld and Arjen P. de Vries. Experimental result analysis for a generative probabilistic image retrieval model. In Callan et al. [27]. ISBN 1-58113-646-3.

[14] Chengxiang Zhai and John Lafferty. 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*, pages 334–342. ACM Press, 2001. ISBN 1-58113-331-6.

[15] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.

[16] Alan F. Smeaton, Wessel Kraaij, and Paul Over. TRECVID 2003 - an introduction. In Alan F. Smeaton, Wessel Kraaij, and Paul Over, editors, *TRECVID 2003 Workshop*, Gaithersburg, MD, USA, 2003. NIST, NIST Special Publications.

[17] Thijs Westerveld, Arjen P. de Vries, Alex van Ballegooij, Fransiska M. G. de Jong, and Djoerd 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.

[18] J-L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Communication*, 37(1–2):89–108, 2002.

[19] Nuno Vasconcelos and Andrew Lippman. Embedded mixture modeling for efficient probabilistic content-based indexing and retrieval. In *Proceedings*

of the SPIE Conference on Multimedia Storage and Archiving Systems III, volume 3527, 1998.

[20] Hayit Greenspan, Jacob Goldberger, and Lenny Ridel. A continuous probabilistic framework for image matching. *Computer Vision and Image Understanding*, 84(3):384–406, 2001. ISSN 1077-3142.

[21] Hangzai Luo, Jianping Fan, Jing Xiao, and Xingquan Zhu. Semantic principal video shot classification via mixture gaussian. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2003.

[22] Hayit Greenspan, Jacob Goldberger, and Arnaldo Mayer. A probabilistic framework for spatio-temporal video representation & indexing. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision*, volume 2353 of *Lecture Notes in Computer Science*, pages 461–475. Springer, 2002.

[23] Hayit Greenspan, Jacob Goldberger, and Arnaldo Mayer. Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, 2004.

[24] Victor Lavrenko. *A generative theory of relevance*. PhD thesis, Graduate school of the Univeristy of Massachusetts Amherst, 2004.

[25] David M. Blei and Michael I. Jordan. Modeling annotated data. In Callan et al. [27]. ISBN 1-58113-646-3.

[26] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In Callan et al. [27]. ISBN 1-58113-646-3.

[27] Jamie Callan, Gordon Cormack, Charles Clarke, David Hawking, and Alan F. Smeaton, editors. *Proceedings of the 26th Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto Canada, 2003. ACM Press. ISBN 1-58113-646-3.