Multimedia Retrieval Using Multiple Examples

Thijs Westerveld and Arjen P. de Vries

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

Abstract. This paper presents a variant of our generative probabilistic multimedia retrieval model. Evaluation on the TRECVID 2003 collection shows the new variant, a document generation approach, is suitable for information needs with multiple examples. Moreover, in combination with textual information, the new variant outperforms the original one.

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 court. 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 [2,3,4]. It is however far from trivial to choose a combination function that works well for a variety of queries.

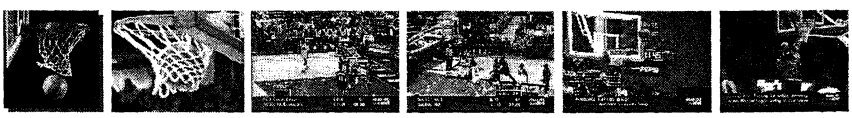


Fig. 1. Topic 101: 'Find shots of a basket being made'.

The present work leaves this approach and captures all the different facets of a set of query examples in a single *topic model*. For retrieval, all documents in a

P. Enser et al. (Eds.): CIVR 2004, LNCS 3115, pp. 344–352, 2004.

collection are compared to this single topic model and ranked accordingly. The 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, $\mathrm{P}(r|D,Q)$, can be estimated indirectly using Bayes' rule: $\mathrm{P}(r|D,Q) = \mathrm{P}(D,Q|r)\mathrm{P}(r)/\mathrm{P}(D,Q)$. For ranking documents, we may avoid estimation of $\mathrm{P}(D,Q)$ using the odds of relevance:

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

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

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

Factoring the conditional probability $\mathrm{P}(D,Q|r)$ 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 $\mathrm{P}(D,Q|r)$ as $\mathrm{P}(D,Q|r) = \mathrm{P}(Q|D,r)\mathrm{P}(D|r)$, giving the following odds of relevance:

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

Since the goal is to rank documents, we can 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:

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

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

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

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

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

3 Generative Multimedia Retrieval

The next step is to define how to estimate the probabilities $\mathrm{P}(Q|D,r)$, $\mathrm{P}(D|Q,r)$ and $\mathrm{P}(D|\bar{r})$. 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.A variant in which temporal aspects are incorporated is presented in [3]. We estimate the (conditional) probabilities of queries and documents, by building a statistical model for each image. Other Generative approaches for multimedia retrieval include [7,8].

3.1 Gaussian Mixture Models

The model assumes that an image is the outcome of a random process that generates n-dimensional feature vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$, 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 $\mathcal{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_{N_S}}\}$.

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 $\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_{N_C}})$ defining the different components. Each component C_i is described by its prior probability $\mathrm{P}(C_i)$, the mean $\boldsymbol{\mu_i}$ and the variance $\boldsymbol{\Sigma_i}$, thus $\boldsymbol{\theta_i} = (\mathrm{P}(C_i), \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$. Details about estimating these parameters are deferred to Section 3.2. Equation 6 defines the probability of drawing one sample $\boldsymbol{x}$ from a GMM with parameters $\boldsymbol{\theta}$.

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{i=1}^{N_C} \mathrm{P}(C_i) \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1} (\boldsymbol{x}-\boldsymbol{\mu_i})} \tag{6}$$

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

$$\mathrm{p}(\mathcal{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N_S} \mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta}) \tag{7}$$

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 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 assignments of samples $\boldsymbol{x_j}$ to components C_i can be viewed as hidden variables, so the Expectation Maximisation (EM) algorithm [9] 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.

$$h_{ij} = \mathrm{P}(C_i|\boldsymbol{x_j}) = \frac{\mathrm{p}(\boldsymbol{x_j}|C_i)\mathrm{P}(C_i)}{\sum_{c=1}^{N_C} \mathrm{p}(\boldsymbol{x_j}|C_c)\mathrm{P}(C_c)} \tag{8}$$

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:

$$\boldsymbol{\mu_i}^{\mathrm{new}} = \frac{\sum_j h_{ij}\boldsymbol{x_j}}{\sum_j h_{ij}}, \tag{9}$$

$$\boldsymbol{\Sigma_i}^{\mathrm{new}} = \frac{\sum_j h_{ij}(\boldsymbol{x_j}-\boldsymbol{\mu_i}^{\mathrm{new}})(\boldsymbol{x_j}-\boldsymbol{\mu_i}^{\mathrm{new}})^T}{\sum_j h_{ij}}, \tag{10}$$

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

The algorithm is guaranteed to converge to a local optimum. In previous experiments we found EM initialisation hardly influences the retrieval results [10].

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 [11]. The estimates of the GMM are smoothed using interpolation with a general, background distribution – this technique is known as Jelinek-Mercer smoothing [12]. The smoothed version of the likelihood for a single sample $\boldsymbol{x}$ becomes (cf. Equation 6):

$$\mathrm{p}_{\mathrm{smooth}}(\boldsymbol{x}|\boldsymbol{\theta}) = \kappa \left[\sum_{i=1}^{N_C} \mathrm{P}(C_i) \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})} \right] + (1-\kappa)\mathrm{p}(\boldsymbol{x}), \tag{12}$$

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 $\mathcal{D}$:

$$\mathrm{p}(\boldsymbol{x}) = \sum_{d \in \mathcal{D}} \mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta_d})\mathrm{P}(d) \tag{13}$$

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 $\boldsymbol{\theta}_D$ (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 13). Thus, the retrieval status values for query generation (Eq. 3) and document generation (Eq. 5) are estimated as

$$\mathrm{RSV}_{\mathrm{Qgen}}(D) = \mathrm{P}(Q|D,r) \equiv \mathrm{P}(\mathcal{X}_Q|\boldsymbol{\theta}_D) \tag{14}$$

$$\mathrm{RSV}_{\mathrm{Dgen}}(D) = \frac{\mathrm{P}(D|Q,r)}{\mathrm{P}(D|\bar{r})} \equiv \frac{\mathrm{P}(\mathcal{X}_D|\boldsymbol{\theta}_Q)}{\mathrm{P}(\mathcal{X}_D)} \tag{15}$$

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 set of query examples, we build an 8-component GMM as described in Section 3.2 ($N_C = 8$). 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.[1] The same sets of samples are used to build topic models $\boldsymbol{\theta}_Q$ for the document generation approach.

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 is evaluated in isolation, as well as in combination with textual information. In the multimodal runs, we use a separate textual model, similar to the query generation approach described before.[2] For each shot a textual model is built from speech transcripts associated with the shot.[3] Assuming independence between the modalities, visual and textual models are used separately, and scores are combined afterwards. For details see [14]. Table 1 shows results for different experimental settings (the last column is explained in Section 4.2). Using full example images, query generation outperforms document generation, but if we select regions, the situation is reversed.

[1] Our manually selected query regions are available from `http://www.cwi.nl/projects/trecvid/trecvid2003/`.

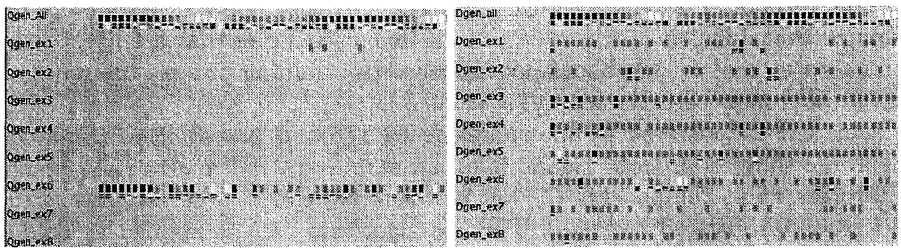
[2] A document generation approach for the textual part is problematic, since the short text queries provide insufficient data to estimate proper topic models from.

[3] The speech transcripts have been kindly provided by LIMSI [13].

Looking at the average precision scores per topic, rather than only at the mean, and inspecting the returned ranked lists for the different models, interesting differences are found. 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, 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. Mean average precision for visual information only, and for a combination with text (MM) (text only: .130). Signs between brackets indicate a significant in/decrease of Dgen in comparison to the corresponding Qgen variant; Wilcoxon signed rank test, at confidence levels of 95% (+/-) and 99% (++/−).

	Qgen		Dgen		Dgen–BG	
Qsamples	visual	MM	visual	MM	visual	MM
full	.028	.143	.026	.119 (-)	.034 (++)	.162 (+)
region	.026	.142	.026 (+)	.167	.034 (++)	.172 (++)



Query Generation Document Generation

Fig. 2. Visualisations of the top 50s for the rocket launch query (topic0107). Each row represents retrieved documents for one run. Documents that are within the top 50 for the multiple-example run (top rows), are assigned a colour code, documents within the top 100 for this run are represented as grey rectangles. If a document from the multiple example run appears in another result, it is represented the same. Documents not in the top 100 for the multiple example run are not represented anywhere. Plots created using NIST's BeadPlot tool (see `http://www.itl.nist.gov/iaui/894.02/projects/beadplot/`).

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 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 an object against a background of sky. The textual information can then help to distinguish between specific objects. Figure 3 shows an example.

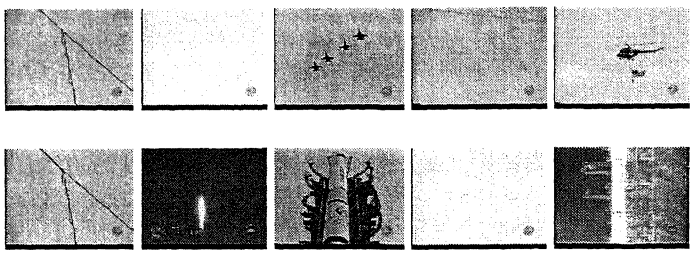


Fig. 3. Document generation results (top 5) for Rocket launch query (topic107). The visual information sets the context (top row, sky background) adding texual information fills in specifics (bottom row, rockets)

4.2 Automatically Selecting Regions

It is clear that selecting regions is useful for the document generation approach. Rather than selecting these 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. A similar approach for text retrieval is studied in [15].

This can be achieved by incorporating background probabilities (Equation 13) in the training process. Again, hidden variables h_{ij} indicate the assignment of samples $\boldsymbol{x_j}$ 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:

$$h_{ij} = \mathrm{P}(C_i|\boldsymbol{x_j}) = \frac{\mathrm{p}(\boldsymbol{x_j}|C_i)\mathrm{P}(C_i)}{\sum_{c=1}^{N_C} \mathrm{p}(\boldsymbol{x_j}|C_c)\mathrm{P}(C_c) + \mathrm{p}(\boldsymbol{x_j})\mathrm{P}(BG)} \tag{16}$$

$$h_{BGj} = \mathrm{P}(BG|\boldsymbol{x_j}) = \frac{\mathrm{p}(\boldsymbol{x_j})\mathrm{P}(BG)}{\sum_{c=1}^{N_C} \mathrm{p}(\boldsymbol{x_j}|C_c)\mathrm{P}(C_c) + \mathrm{p}(\boldsymbol{x_j})\mathrm{P}(BG)}, \tag{17}$$

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

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

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

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 1 shows the results for the new EM variant. A small (1%) sample from a comparable collection (the TRECVID 2003 development set) was used to estimate the background probabilities ($P(\boldsymbol{x}_j)$) 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, only 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 captures mainly things that are present in all examples, and thus builds a model that describes the commonalities shared by 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. Our automatic approach yields results comparable to manual region selection. Automatically selecting important parts within manually created regions gives another improvement (though slight) on the scores over using the user's manual selection as is.

Future work on the document generation model should prove whether 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. In: TRECVID 2003 Workshop. (2003)
2. Jin, X., French, J.C.: Improving image retrieval effectiveness via multiple queries. In: Proceedings of the first ACM int. workshop on Multimedia databases. (2003)
3. Westerveld, T., Ianeva, T., Boldareva, L., de Vries, A.P., Hiemstra, D.: Combining information sources for video retrieval. In: TRECVID 2003 Workshop. (2003)
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. IP&M **36** (2000)
6. Lafferty, J., Zhai, C.: Probabilistic IR models based on document and query generation. In: Language Modeling for Information Retrieval. (2003)
7. Blei, D.M., Jordan, M.I.: Modeling annotated data. In: ACM SIGIR 2003. (2003)
8. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: ACM SIGIR 2003. (2003)
9. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journ. of the Royal Statistical Society, series B **39** (1977)
10. Westerveld, T., de Vries, A.P.: Experimental result analysis for a generative probabilistic image retrieval model. In: ACM SIGIR 2003. (2003)
11. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: ACM SIGIR 2001. (2001)
12. Jelinek, F., Mercer, R.: Interpolated estimation of markov source parameters from sparse data. In: Proc. of the Workshop on Pattern Recognition in Practice. (1980)
13. Gauvain, J., Lamel, L., Adda, G.: The LIMSI broadcast news transcription system. Speech Communication **37** (2002)
14. 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)
15. Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. In: ACM SIGIR 2004. (2004)